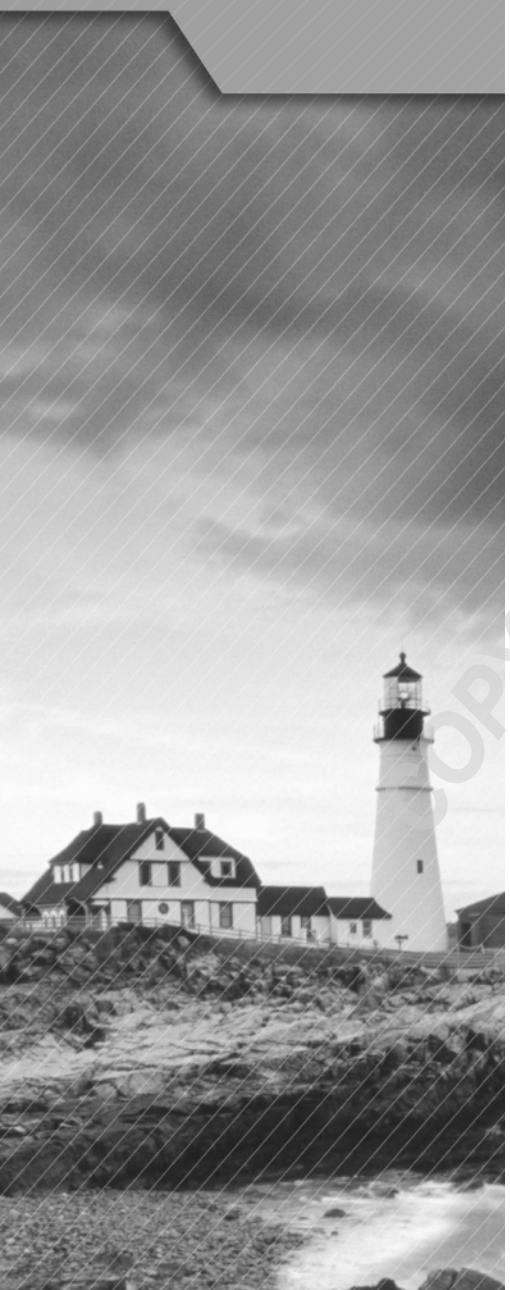
Chapter

1

Operation of IP Data Networks



THE FOLLOWING CCNA ROUTING AND SWITCHING EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

- ✓ Operation of IP Data Networks
- ✓ Recognize the purpose and functions of various network devices such as Routers, Switches, Bridges and Hubs.
- ✓ Select the components required to meet a given network specification.
- ✓ Identify common applications and their impact on the network.
- ✓ Describe the purpose and basic operation of the protocols in the OSI and TCP/IP models.
- ✓ Predict the data flow between two hosts across a network.
- ✓ Identify the appropriate media, cables, ports, and connectors to connect Cisco network devices to other network devices and hosts in a LAN.

In this chapter, I will review the basics of internetworking and what an internetwork is. I will go over some of the components that make up a network as well as some applications used in networking. I will also go over the OSI and TCP/IP models and, finally, explain how data flows across a network as well as discuss the various connectors used in a network.

Operation of IP Data Networks

Let's start by defining exactly what an internetwork is: You create an internetwork when you connect two or more networks via a router and configure a logical network addressing scheme with a protocol such as IPv4 or IPv6.

Why is it so important to learn Cisco internetworking anyway? Networks and networking have grown exponentially over the past 20 years, and understandably so. They've had to evolve at light speed just to keep up with huge increases in basic, mission-critical user needs (for example, simple sharing of data and printers) as well as greater burdens like multimedia remote presentations and conferencing. Unless everyone who needs to share network resources is located in the same office space—an increasingly uncommon situation—the challenge is to connect relevant networks so all users can share the wealth of whatever services and resources are required. Figure 1.1 shows a basic *local area network (LAN)* that's connected using a *hub*, which is basically just an antiquated device that connects wires together. Keep in mind that a simple network like this would be considered one collision domain and one broadcast domain.

FIGURE 1.1 A very basic network

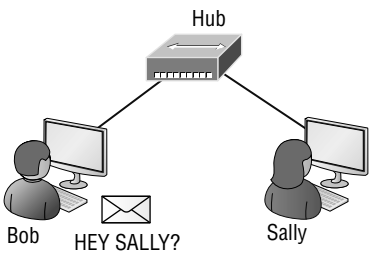


Figure 1.1 illustrates this scenario: Bob wants to send Sally a file, and to complete that goal in this kind of network, he'll simply broadcast that he's looking for her, which is basically just shouting out over the network. As networks grow and get more complex, a good network design is essential. Growth can be good, but growth can also hinder your network. LAN congestion can bring your network to a halt! The solution to this is to break up a large network into smaller networks, which is called *network segmentation*.

This concept is a lot like planning a new community or modernizing an existing one. More streets are added, complete with new intersections and traffic signals, plus post offices with official maps documenting all those street names and directions on how to get to each are built. You'll need to effect new laws to keep order to it all and provide a police station to protect this nice new neighborhood as well. In a networking neighborhood environment, all of this is carried out using devices like *routers*, *switches*, and *bridges*.

Exam Essentials

Understand what an internetwork is. An internetwork consists of two or more networks that are connected together via a router. Networks are configured with a logical addressing schemes and segmented into smaller networks using routers, switches, and bridges.

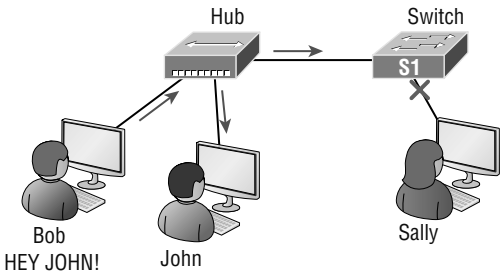
Recognize the Purpose and Functions of Various Network Devices Such as Routers, Switches, Bridges, and Hubs

The scenario I just described brings me to the basic point of what this book and the Cisco certification objectives are really all about. My goal of showing you how to create efficient networks and segment them correctly in order to minimize all the chaotic yelling and screaming going on in them is a universal theme throughout my CCENT and CCNA series books. It's just inevitable that you'll have to break up a large network into a bunch of smaller ones at some point to match a network's equally inevitable growth, and as that expansion occurs, user response time simultaneously dwindles to a frustrating crawl. But if you master the vital technology and skills I have in store for you in this book, you'll be well equipped to rescue your network and its users by creating an efficient new network neighborhood to give them key amenities like the bandwidth they need to meet their evolving demands.

And this is no joke; most of us think of growth as good—and it can be—but as many of us experience daily when commuting to work, school, etc., it can also mean your LAN's traffic congestion can reach critical mass and grind to a complete halt!

So let's take a look at our new neighborhood now, because the word has gotten out; many more hosts have moved into it, so it's time to upgrade that new high-capacity infrastructure that we promised to handle the increase in population. Figure 1.2 shows a network that's been segmented with a switch, making each network segment that connects to the switch its own separate collision domain. Doing this results in a lot less yelling!

FIGURE 1.2 A switch can break up collision domains.



This is still one single broadcast domain. You can see that the hub used in Figure 1.2 just extended the one collision domain from the switch port. The result is that John received the data from Bob but, happily, Sally did not. This is good because Bob intended to talk with John directly, and if he had needed to send a broadcast instead, everyone, including Sally, would have received it, possibly causing unnecessary congestion.

Here's a list of some of the things that commonly cause LAN traffic congestion:

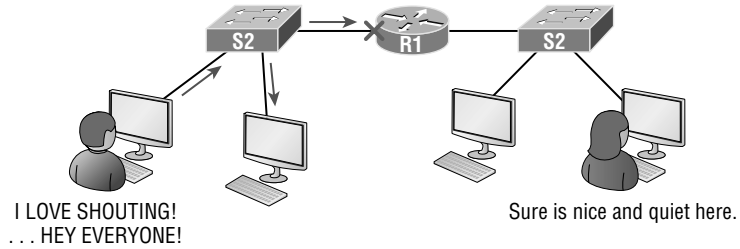
- Too many hosts in a collision or broadcast domain
- Broadcast storms
- Too much multicast traffic
- Low bandwidth
- Adding hubs for connectivity to the network
- A bunch of ARP broadcasts

Take another look at Figure 1.2 and make sure you see that I extended the main hub from Figure 1.1 to a switch in Figure 1.2. I did that because hubs don't segment a network; they just connect network segments.

As a network begins to grow, routers are used to connect networks and route packets of data from one network to another. Cisco became the de facto standard for routers because of its unparalleled selection of high-quality router products and fantastic service. So never forget that by default, routers are basically employed to efficiently break up a *broadcast domain*—the set of all devices on a network segment, which are allowed to "hear" all broadcasts sent out on that specific segment.

Figure 1.3 depicts a router in our growing network, creating an internetwork and breaking up broadcast domains.

FIGURE 1.3 Routers create an internetwork.



Each host is connected to its own collision domain because of the switch, and the router has created two broadcast domains. Routers also provide connections to wide area network (WAN) services as well as via a serial interface for WAN connections—specifically, a V.35 physical interface on a Cisco router.

Even though routers are known for breaking up broadcast domains by default, it's important to remember that they break up collision domains as well.

There are two advantages to using routers in your network:

- They don't forward broadcasts by default.
- They can filter the network based on layer 3, Network layer, information such as an IP address.

Here are four ways a router functions in your network:

- Packet switching
- Packet filtering
- Internetwork communication
- Path selection

It's helpful to think of routers as layer 3 switches. Unlike layer 2 switches, which forward or filter frames, routers (layer 3 switches) use logical addressing and provide an important capacity called *packet switching*. Routers can also provide packet filtering via access lists, and when routers connect two or more networks together and use logical addressing (IPv4 or IPv6), you then have an *internetwork*. Finally, routers use a routing table, which is essentially a map of the internetwork, to make best path selections for getting data to its proper destination and properly forward packets to remote networks.

Conversely, we don't use layer 2 switches to create internetworks because they don't break up broadcast domains by default. Instead, they're employed to add functionality to a network LAN. The main purpose of these switches is to make a LAN work better—to optimize its performance—providing more bandwidth for the LAN's users. Also, these switches don't forward packets to other networks like routers do. Instead, they only "switch" frames from one port to another within the switched network.

By default, switches break up collision domains, but what are these things? *Collision domain* is an Ethernet term used to describe a network scenario in which one device sends a packet out on a network segment and every other device on that same segment is forced to pay attention no matter what. This isn't very efficient because if a different device tries to transmit at the same time, a collision will occur, requiring both devices to retransmit, one at a time—not good! This happens a lot in a hub environment, where each host segment connects to a hub that represents only one collision domain and a single broadcast domain. By contrast, each and every port on a switch represents its own collision domain, allowing network traffic to flow much more smoothly.

Switches create separate collision domains within a single broadcast domain. Routers provide a separate broadcast domain for each interface. Don't let this ever confuse you!

You'll still hear Cisco and others refer to LAN switches as multiport bridges now and then.

Basically, switches are multiport bridges with more brain power and more ports!

You would use a bridge in a network to reduce collisions within broadcast domains and to increase the number of collision domains in your network. Doing this provides more bandwidth for users. And never forget that using hubs in your Ethernet network can contribute to congestion. As always, plan your network design carefully!

Exam Essentials

Describe the difference between a collision domain and a broadcast domain. *Collision domain* is an Ethernet term used to describe a network collection of devices in which one particular device sends a packet on a network segment, forcing every other device on that same segment to pay attention to it. With a broadcast domain, a set of all devices on a network hears all broadcasts sent on all segments.

Understand the difference between a hub, a bridge/switch, and a router. All ports on a hub are in one collision domain. When data is received on a port, it is sent out to all ports simultaneously. Each port on a switch is a separate collision domain. When data is received on a switchport, it is sent only to the receiving host that needs it. Routers are used to create internetworks and provide connections to WAN services. Routers break up broadcast and collision domains as well.

Select the Components Required to Meet a Given Network Specification

The term *bridging* was introduced before routers and switches were implemented, so it's pretty common to hear people referring to switches as bridges. That's because bridges and switches basically do the same thing—break up collision domains on a LAN. Note to self that you cannot buy a physical bridge these days, only LAN switches, which use bridging technologies. This means that you'll still hear Cisco and others refer to LAN switches as multiport bridges now and then.

But does it mean that a switch is just a multiple-port bridge with more brainpower? Well, pretty much, only there are still some key differences. Switches do provide a bridging function, but they do that with greatly enhanced management ability and features. Plus, most bridges had only 2 or 4 ports, which is severely limiting. Of course, it was possible to get your hands on a bridge with up to 16 ports, but that's nothing compared to the hundreds of ports available on some switches!

Figure 1.4 shows how a network would look with various internetwork devices in place. Remember, a router doesn't just break up broadcast domains for every LAN interface; it breaks up collision domains too.

Looking at Figure 1.4, did you notice that the router has the center stage position and connects each physical network together? I'm stuck with using this layout because of the ancient bridges and hubs involved. I really hope you don't run across a network like this, but it's still really important to understand the strategic ideas that this figure represents!

See that bridge up at the top of the internetwork shown in Figure 1.4? It's there to connect the hubs to a router. The bridge breaks up collision domains, but all the hosts connected to both hubs are still crammed into the same broadcast domain. That bridge also created only three collision domains, one for each port, which means that each device connected to a hub is in the same collision domain as every other device connected to that same hub. This is really lame and to be avoided if possible, but it's still better than having one collision domain for all hosts! So don't do this at home; it's a great museum piece and a wonderful example of what not to do, but this inefficient design would be terrible for use in today's networks! It does show us how far we've come though, and again, the foundational concepts it illustrates are really important for you to get.

The three interconnected hubs at the bottom of the figure also connect to the router. This setup creates one collision domain and one broadcast domain and makes that bridged network, with its two collision domains, look much better by contrast!

The best network connected to the router is the LAN switched network on the left. Why? Because each port on that switch breaks up collision domains. But it's not all good—all devices are still in the same broadcast domain. This can be bad because all devices must listen to all broadcasts transmitted. And if your broadcast domains are too large, the users have less bandwidth and are required to process more broadcasts. Network response time eventually will slow to a level that could cause your users to riot and strike, so it's important to keep your broadcast domains small in the vast majority of networks today.

FIGURE 1.4 Internetworking devices

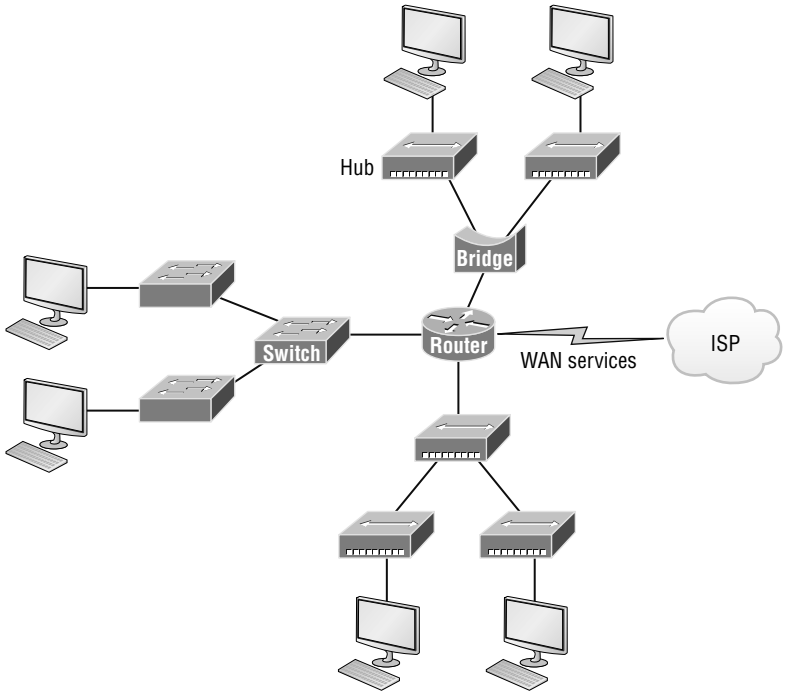
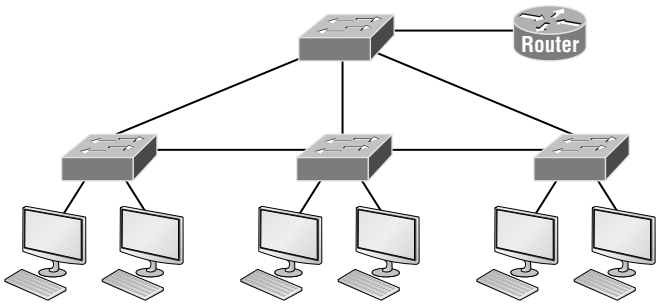


Figure 1.5 demonstrates a network you'll typically stumble upon today.

LAN switches are at the center of this network, with the routers connecting the logical networks. If I went ahead and implemented this design, I've created something called virtual LANs, or VLANs, which are used when you logically break up broadcast domains in a layer 2 switched network. It's really important to understand that even in a switched network environment, you still need a router to provide communication between VLANs.

FIGURE 1.5 Switched networks creating an internetwork



Still, clearly the best network design is the one that's perfectly configured to meet the business requirements of the specific company or client it serves, and it's usually one in which LAN switches exist in harmony with routers strategically placed in the network.

Let's look at Figure 1.4 again. How many collision domains and broadcast domains are really there in this internetwork?

The all-hub network at the bottom is one collision domain; the bridge network on top equals three collision domains. Add in the switch network of five collision domains—one for each switch port—and you get a total of nine!

In Figure 1.5, each port on the switch is a separate collision domain, and each VLAN would be a separate broadcast domain. So how many collision domains do you see here? I'm counting 12—remember that connections between the switches are considered a collision domain! Since the figure doesn't show any VLAN information, we can assume that the default of one VLAN, or one broadcast domain, is in place.

Exam Essentials

Understand the importance of essential network design. Placing routers and switches in a properly designed network configuration will fulfill the needs of a specific company or client and will operate with optimal performance.

Identify the functions and advantages of routers. Routers perform packet switching, filtering, and path selection, and they facilitate internetwork communication. One advantage of routers is that they reduce broadcast traffic.

Identify Common Applications and Their Impact on the Network

In this section, we'll go over the different applications and services typically used in IP networks, and although there are many more protocols defined here, we'll focus on the protocols most relevant to the CCNA objectives. Here's a list of the protocols and applications we'll cover in this section:

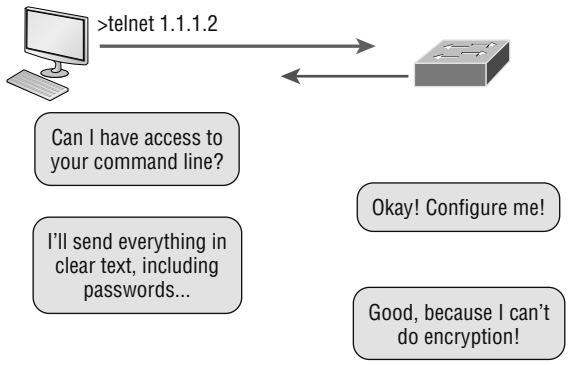
- Telnet
- SSH
- FTP
- TFTP
- SNMP
- HTTP
- HTTPS

- NTP
- DNS
- DHCP/BootP

Telnet

Telnet was one of the first Internet standards, developed in 1969, and is the chameleon of protocols—its specialty is terminal emulation. It allows a user on a remote client machine, called the Telnet client, to access the resources of another machine, the Telnet server, in order to access a command-line interface. Telnet achieves this by pulling a fast one on the Telnet server and making the client machine appear as though it were a terminal directly attached to the local network. This projection is actually a software image—a virtual terminal that can interact with the chosen remote host. A drawback is that there are no encryption techniques available within the Telnet protocol, so everything must be sent in clear text, including passwords! Figure 1.6 shows an example of a Telnet client trying to connect to a Telnet server.

FIGURE 1.6 Telnet

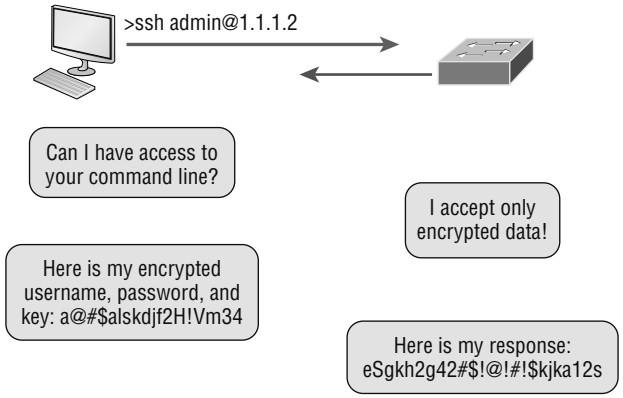


These emulated terminals are of the text-mode type and can execute defined procedures such as displaying menus that give users the opportunity to choose options and access the applications on the duped server. Users begin a Telnet session by running the Telnet client software and then logging into the Telnet server. Telnet uses an 8-bit, byte-oriented data connection over TCP, which makes it very thorough. It's still in use today because it is so simple and easy to use, with very low overhead, but again, with everything sent in clear text, it's not recommended in production.

Secure Shell (SSH)

Secure Shell (SSH) protocol sets up a secure session that's similar to Telnet over a standard TCP/IP connection and is employed for doing things like logging into systems, running programs on remote systems, and moving files from one system to another. And it does all of this while maintaining an encrypted connection. Figure 1.7 shows an SSH client trying to connect to an SSH server. The client must send the data encrypted!

FIGURE 1.7 Secure Shell



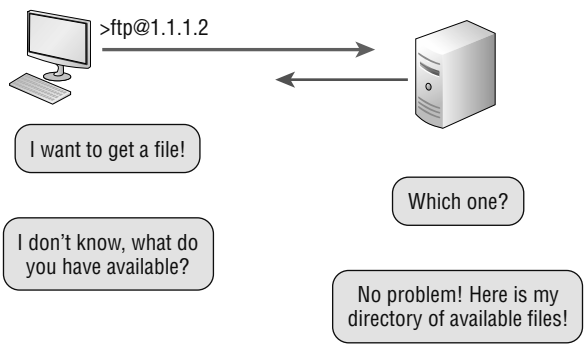
You can think of it as the new-generation protocol that's now used in place of the antiquated and very unused `rsh` and `rlogin`—even Telnet.

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) actually lets us transfer files, and it can accomplish this between any two machines using it. But FTP isn't just a protocol; it's also a program. Operating as a protocol, FTP is used by applications. As a program, it's employed by users to perform file tasks by hand. FTP also allows for access to both directories and files and can accomplish certain types of directory operations, such as relocating into different ones (Figure 1.8).

But accessing a host through FTP is only the first step. Users must then be subjected to an authentication login that's usually secured with passwords and usernames implemented by system administrators to restrict access. You can get around this somewhat by adopting the username *anonymous*, but you'll be limited in what you'll be able to access.

FIGURE 1.8 FTP



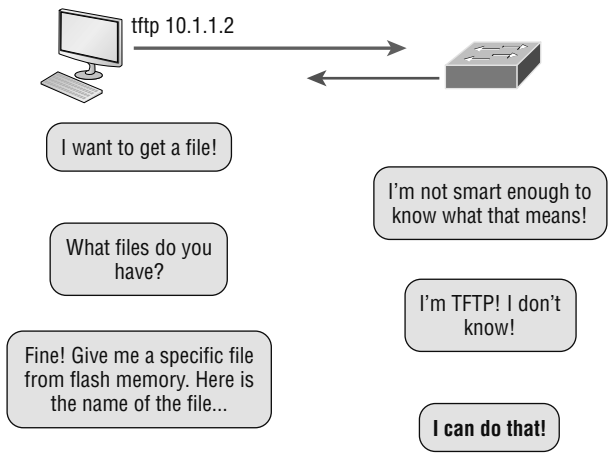
Even when employed by users manually as a program, FTP's functions are limited to listing and manipulating directories, typing file contents, and copying files between hosts. It can't execute remote files as programs.

Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP) is the stripped-down, stock version of FTP, but it's the protocol of choice if you know exactly what you want and where to find it because it's fast and so easy to use!

But TFTP doesn't offer the abundance of functions that FTP does because it has no directory-browsing abilities, meaning that it can only send and receive files (Figure 1.9).

FIGURE 1.9 TFTP



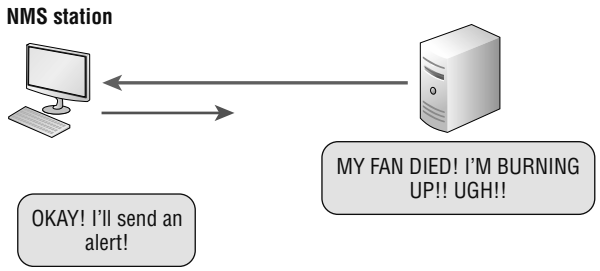
This compact little protocol also skimps in the data department, sending much smaller blocks of data than FTP. Also, there's no authentication as with FTP, so it's even more insecure, and few sites support it because of the inherent security risks.

Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) collects and manipulates valuable network information, as you can see in Figure 1.10. It gathers data by polling the devices on the network from a network management station (NMS) at fixed or random intervals, requiring them to disclose certain information, or even by asking for certain information from the device. In addition, network devices can inform the NMS station about problems as they occur so the network administrator is alerted.

When all is well, SNMP receives something called a *baseline*—a report delimiting the operational traits of a healthy network. This protocol can also stand as a watchdog over the network, quickly notifying managers of any sudden turn of events. These network watchdogs are called *agents*, and when aberrations occur, agents send an alert called a *trap* to the management station.

FIGURE 1.10 SNMP

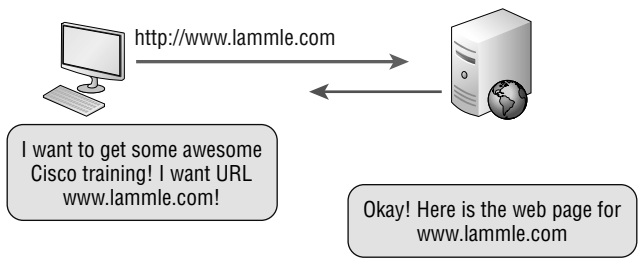


Hypertext Transfer Protocol (HTTP)

All those snappy websites comprising a mélange of graphics, text, links, ads, and so on, rely on the *Hypertext Transfer Protocol (HTTP)* to make it all possible (Figure 1.11). It's used to manage communications between web browsers and web servers and opens the right resource when you click a link, wherever that resource may actually reside.

In order for a browser to display a web page, it must find the exact server that has the right web page, plus the exact details that identify the information requested. This information must then be sent back to the browser. Nowadays, it's highly doubtful that a web server would have only one page to display!

FIGURE 1.11 HTTP



Your browser can understand what you need when you enter a Uniform Resource Locator (URL), which we usually refer to as a web address, such as, for example, `www.lammle.com/forum` and `www.lammle.com/blog`.

So basically, each URL defines the protocol used to transfer data, the name of the server, and the particular web page on that server.

Hypertext Transfer Protocol Secure (HTTPS)

Hypertext Transfer Protocol Secure (HTTPS) is also known as Secure Hypertext Transfer Protocol. It uses Secure Sockets Layer (SSL) or Transport Layer Security (TLS). Sometimes you'll see it referred to as SHTTP or S-HTTP, which were slightly different protocols, but

since Microsoft supported HTTPS, it became the de facto standard for securing web communication. But no matter—as indicated, it's a secure version of HTTP that arms you with a whole bunch of security tools for keeping transactions between a web browser and a server secure.

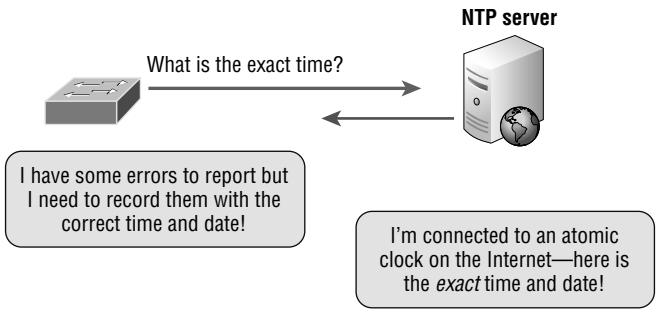
It's what your browser needs to fill out forms, sign in, authenticate, and encrypt an HTTP message when you do things online like make a reservation, access your bank account, or buy something.

Network Time Protocol (NTP)

Kudos to professor David Mills of the University of Delaware for coming up with this handy protocol that's used to synchronize the clocks on our computers to one standard time source (typically, an atomic clock). *Network Time Protocol (NTP)* works by synchronizing devices to ensure that all computers on a given network agree on the time (Figure 1.12).

This may sound pretty simple, but it's very important because so many of the transactions done today are time and date stamped. Think about databases—a server can get messed up pretty badly and even crash if it's out of sync with the machines connected to it by even mere seconds! You can't have a transaction entered by a machine at, say, 1:50 a.m. when the server records that transaction as having occurred at 1:45 a.m. So basically, NTP works to prevent a "back to the future *sans* DeLorean" scenario from bringing down the network—very important indeed!

FIGURE 1.12 NTP



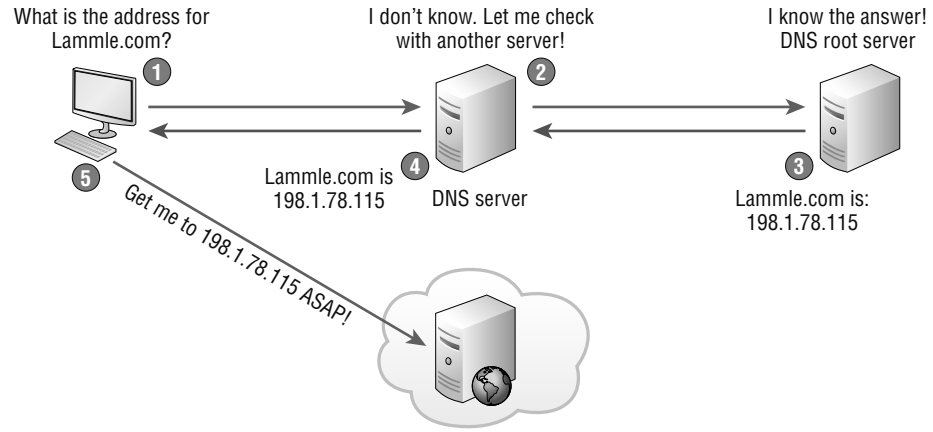
Domain Name Service (DNS)

Domain Name Service (DNS) resolves hostnames—specifically, Internet names, such as `www.lammle.com`. But you don't have to actually use DNS. You just type in the IP address of any device you want to communicate with and find the IP address of a URL by using the Ping program. For example, `>ping www.cisco.com` will return the IP address resolved by DNS.

An IP address identifies hosts on a network and the Internet as well, but DNS was designed to make our lives easier. Think about this: What would happen if you wanted to move your web page to a different service provider? The IP address would change and no one would know what the new one was. DNS allows you to use a domain name to specify an IP address. You can change the IP address as often as you want and no one will know the difference.

To resolve a DNS address from a host, you'd typically type in the URL from your favorite browser, which would hand the data to the Application layer interface to be transmitted on the network. The application would look up the DNS address and send a UDP request to your DNS server to resolve the name (Figure 1.13).

FIGURE 1.13 DNS



If your first DNS server doesn't know the answer to the query, then the DNS server forwards a TCP request to its root DNS server. Once the query is resolved, the answer is transmitted back to the originating host, which means the host can now request the information from the correct web server.

DNS is used to resolve a *fully qualified domain name (FQDN)*—for example, `www.lammle.com` or `todd.lammle.com`. An FQDN is a hierarchy that can logically locate a system based on its domain identifier.

If you want to resolve the name *todd*, you must either type in the FQDN of `todd.lammle.com` or have a device such as a PC or router add the suffix for you. For example, on a Cisco router, you can use the command `ip domain-name lammle.com` to append each request with the `lammle.com` domain. If you don't do that, you'll have to type in the FQDN to get DNS to resolve the name.

An important thing to remember about DNS is that if you can ping a device with an IP address but cannot use its FQDN, then you might have some type of DNS configuration failure.

Dynamic Host Configuration Protocol (DHCP)/Bootstrap Protocol (BootP)

Dynamic Host Configuration Protocol (DHCP) assigns IP addresses to hosts. It allows for easier administration and works well in small to very large network environments. Many types of hardware can be used as a DHCP server, including a Cisco router.

DHCP differs from BootP in that BootP assigns an IP address to a host but the host's hardware address must first be entered manually in a BootP table on the BootP server. You can think of DHCP as a dynamic BootP. But remember that BootP is also used to send an operating system to a host and the host can boot from it. DHCP can't do that.

But there's still a lot of information a DHCP server can provide to a host when the host is requesting an IP address from the DHCP server. Here's a list of the most common types of information a DHCP server can provide:

- IP address
- Subnet mask
- Domain name
- Default gateway (routers)
- DNS server address
- WINS server address

A client that sends out a DHCP Discover message in order to receive an IP address sends out a broadcast at both layer 2 and layer 3:

- The layer 2 broadcast is all *f*s in hex, which looks like this: ff:ff:ff:ff:ff:ff.
- The layer 3 broadcast is 255.255.255.255, which means all networks and all hosts.

DHCP is connectionless, which means it uses User Datagram Protocol (UDP) at the Transport layer, also known as the Host-to-Host layer.

Seeing is believing, so here's an example of output from my analyzer showing the layer 2 and layer 3 broadcasts:

```
Ethernet II, Src: 0.0.0.0 (00:0b:db:99:d3:5e),Dst: Broadcast(ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0),Dst: 255.255.255.255(255.255.255.255)
```

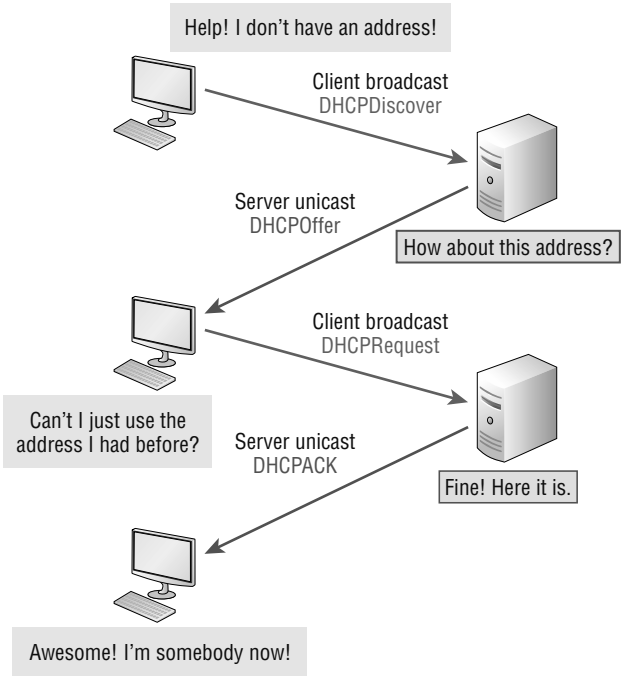
The Data Link and Network layers are both sending out "all hands" broadcasts saying, "Help—I don't know my IP address!"

Figure 1.14 shows the process of a client-server relationship using a DHCP connection.

This is the four-step process a client takes to receive an IP address from a DHCP server:

1. The DHCP client broadcasts a DHCP Discover message looking for a DHCP server (UDP port 67).
2. The DHCP server that received the DHCP Discover message sends a layer 2 unicast DHCP Offer (UDP port 68) message back to the host.
3. The client then broadcasts to the server a DHCP Request message asking for the offered IP address and possibly other information.
4. The server finalizes the exchange with a unicast DHCP Acknowledgment message.

FIGURE 1.14 DHCP client four-step process



DHCP Conflicts

A DHCP address conflict occurs when two hosts use the same IP address. This sounds bad, and it is!

During IP address assignment, a DHCP server checks for conflicts using the Ping program to test the availability of the address before it's assigned from the pool. If no host replies, then the DHCP server assumes that the IP address is not already allocated. This helps the server know that it's providing a good address, but what about the host? To provide extra protection against that terrible IP conflict issue, the host can broadcast for its own address!

A host uses something called a gratuitous ARP to help avoid a possible duplicate address. A gratuitous ARP is an ARP response made to all devices on the network when there was never an original ARP request. Using this gratuitous ARP, the DHCP client sends an ARP broadcast out on the local LAN or VLAN using its newly assigned address to solve conflicts before they occur.

So, if an IP address conflict is detected, the address is removed from the DHCP pool (scope), and it's really important to remember that the address will not be assigned to a host until the administrator resolves the conflict by hand!

Exam Essentials

Identify Process/Application layer protocols. Telnet is a terminal emulation program that allows you to log into a remote host and run programs. File Transfer Protocol (FTP) is a connection-oriented service that allows you to transfer files. Trivial FTP (TFTP) is a connectionless file transfer program. Simple Mail Transfer Protocol (SMTP) is a sendmail program.

Describe the functions of DNS and DHCP in the network. Dynamic Host Configuration Protocol (DHCP) provides network configuration information (including IP addresses) to hosts, eliminating the need to perform the configurations manually. Domain Name Service (DNS) resolves hostnames—both Internet names such as `www.lammle.com` and device names such as Workstation 2—to IP addresses, eliminating the need to know the IP address of a device for connection purposes.

Describe the Purpose and Basic Operation of the Protocols in the OSI and TCP/IP Models

When networks first came into being, computers could typically communicate only with computers from the same manufacturer. For example, companies ran either a complete DECnet solution or an IBM solution, never both together. In the late 1970s, the *Open Systems Interconnection (OSI) reference model* was created by the International Organization for Standardization (ISO) to break through this barrier.

The OSI model was meant to help vendors create interoperable network devices and software in the form of protocols so that different vendor networks could work in peaceable accord with each other.

The Layered Approach

A *reference model* is a conceptual blueprint of how communications should take place. It addresses all the processes required for effective communication and divides them into logical groupings called *layers*. When a communication system is designed in this manner, it's known as a *layered architecture* because it's hierarchical.

Models happen to be really important to software developers too. They often use a reference model to understand computer communication processes so they can determine which functions should be accomplished on a given layer. This means that if someone is creating a protocol for a certain layer, they need to be concerned only with their target layer's function. Software that maps to another layers' protocols and is specifically designed to be deployed there will handle additional functions. The technical term for this idea is *binding*. The communication processes that are related to each other are bound, or grouped together, at a particular layer.

Advantages of Reference Models

The OSI model is hierarchical, and there are many advantages that can be applied to any layered model, but as I said, the OSI model's primary purpose is to allow different vendors' networks to interoperate.

Here's a list of some of the more important benefits for using the OSI layered model:

- It divides the network communication process into smaller and simpler components, facilitating component development, design, and troubleshooting.
- It allows multiple-vendor development through the standardization of network components.
- It encourages industry standardization by clearly defining what functions occur at each layer of the model.
- It allows various types of network hardware and software to communicate.
- It prevents changes in one layer from affecting other layers to expedite development.
- It eases the learning process by allowing you to understand the functions, benefits, and considerations of one layer at a time instead of having to overcome one large and complex subject.

The OSI Reference Model

One of best gifts the OSI specifications gives us is paving the way for the data transfer between disparate hosts running different operating systems, like Unix hosts, Windows machines, Macs, smartphones, and so on.

The OSI is a logical model, not a physical one. It's essentially a set of guidelines that developers can use to create and implement applications to run on a network. It also provides a framework for creating and implementing networking standards, devices, and internetworking schemes.

The OSI has seven different layers, divided into two groups. The top three layers (known as the upper layers) define how the applications within the end stations will communicate with each other as well as with users. The bottom four layers define how data is transmitted end to end.

Figure 1.15 shows the three upper layers and their functions.

FIGURE 1.15 The upper layers

Application	• Provides a user interface
Presentation	• Presents data • Handles processing such as encryption
Session	• Keeps different applications' data separate

When looking at Figure 1.15, understand that users interact with the computer at the Application layer and also that the upper layers are responsible for applications communicating between hosts. None of the upper layers knows anything about networking or network addresses because that's the responsibility of the four bottom layers.

Figure 1.16 shows the four lower layers and their functions. You can see that it's these four bottom layers that define how data is transferred through physical media like wire, cable, fiber optics, switches, and routers. These bottom layers also determine how to rebuild a data stream from a transmitting host to a destination host's application.

FIGURE 1.16 The lower layers

Transport	• Provides reliable or unreliable delivery • Performs error correction before retransmit
Network	• Provides logical addressing, which routers use for path determination
Data Link	• Combines packets into bytes and bytes into frames • Provides access to media using MAC address • Performs error detection not correction
Physical	• Moves bits between devices • Specifies voltage, wire speed, and pin-out of cables

The following network devices and protocols operate at all seven layers of the OSI model:

- Network management stations (NMSs)
- Web and application servers
- Gateways (not default gateways)
- Servers
- Network hosts

Basically, the ISO is pretty much the Emily Post of the network protocol world. Just as Ms. Post wrote the book setting the standards—or protocols—for human social interaction, the ISO developed the OSI reference model as the precedent and guide for an open network protocol set. Defining the etiquette of communication models, it remains the most popular means of comparison for protocol suites today.

As you've just seen, the OSI reference model has the following seven layers:

- Application layer (layer 7)
- Presentation layer (layer 6)
- Session layer (layer 5)
- Transport layer (layer 4)
- Network layer (layer 3)
- Data Link layer (layer 2)
- Physical layer (layer 1)

Some people like to use a mnemonic to remember the seven layers, such as **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing. Figure 1.17 shows a summary of the functions defined at each layer of the OSI model.

FIGURE 1.17 OSI layer functions

Application	• File, print, message, database, and application services
Presentation	• Data encryption, compression, and translation services
Session	• Dialog control

Transport	• End-to-end connection
Network	• Routing

Data Link	• Framing
Physical	• Physical topology

I've separated the seven-layer model into three different functions: the upper layers, the middle layers, and the bottom layers. The upper layers communicate with the user interface and application, the middle layers do reliable communication and routing to a remote network, and the bottom layers communicate to the local network.

With this in hand, you're now ready to explore each layer's function in detail!

The Application Layer

The *Application layer* of the OSI model marks the spot where users actually communicate to the computer and comes into play only when it's clear that access to the network will be needed soon. Take the case of Internet Explorer (IE). You could actually uninstall every trace of networking components like TCP/IP, the NIC card, and so on, and still use IE to view a local HTML document. But things would get ugly if you tried to do things like view a remote HTML document that must be retrieved because IE and other browsers act on these types of requests by attempting to access the Application layer. So basically, the Application layer is working as the interface between the actual application program and the next layer down by providing ways for the application to send information down through the protocol stack. This isn't actually part of the layered structure because browsers don't live in the Application layer, but they interface with it as well as the relevant protocols when asked to access remote resources.

The Presentation Layer

The *Presentation layer* gets its name from its purpose: It presents data to the Application layer and is responsible for data translation and code formatting. Think of it as the OSI model's translator, providing coding and conversion services. One very effective way of

ensuring a successful data transfer is to convert the data into a standard format before transmission. Computers are configured to receive this generically formatted data and then reformat it back into its native state to read it. An example of this type of translation service occurs when translating old IBM Extended Binary Coded Decimal Interchange Code (EBCDIC) data to PC ASCII, the American Standard Code for Information Interchange (often pronounced "askee"). So just remember that by providing translation services, the Presentation layer ensures that data transferred from the Application layer of one system can be read by the Application layer of another one.

The Session Layer

The *Session layer* is responsible for setting up, managing, and dismantling sessions between Presentation layer entities and keeping user data separate. Dialog control between devices also occurs at this layer.

The Transport Layer

The *Transport layer* segments and reassembles data into a single data stream. Services located at this layer take all the various data received from upper-layer applications and then combine it into the same, concise data stream. These protocols provide end-to-end data transport services and can establish a logical connection between the sending host and destination host on an internetwork.

A pair of well-known protocols called TCP and UDP are integral to this layer, and understand that although both work at the Transport layer, TCP is known as a reliable service but UDP is not. This distinction gives application developers more options because they have a choice between the two protocols when they are designing products for this layer.

The Transport layer is responsible for providing mechanisms for multiplexing upper-layer applications, establishing sessions, and tearing down virtual circuits. It can also hide the details of network-dependent information from the higher layers as well as provide transparent data transfer.

The term *reliable networking* can be used at the Transport layer. Reliable networking requires that acknowledgments, sequencing, and flow control will all be used.

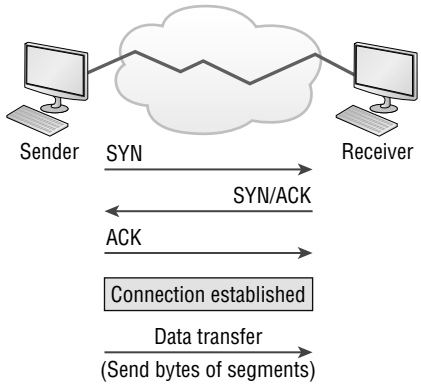
The Transport layer can be either connectionless or connection-oriented, but because Cisco really wants you to understand the connection-oriented function of the Transport layer, I'm going to go into that in more detail here.

Connection-Oriented Communication

For reliable transport to occur, a device that wants to transmit must first establish a connection-oriented communication session with a remote device (its peer system), known as a *call setup* or a *three-way handshake*. Once this process is complete, the data transfer occurs, and when it's finished, a call termination takes place to tear down the virtual circuit.

Figure 1.18 depicts a typical reliable session taking place between sending and receiving systems. In it, you can see that both hosts' application programs begin by notifying their individual operating systems that a connection is about to be initiated. The two operating systems communicate by sending messages over the network confirming that the transfer is approved and that both sides are ready for it to take place. After all of this required synchronization takes place, a connection is fully established and the data transfer begins. And by the way, it's really helpful to understand that this virtual circuit setup is often referred to as overhead!

FIGURE 1.18 Establishing a connection-oriented session



While the information is being transferred between hosts, the two machines periodically check in with each other, communicating through their protocol software to ensure that all is going well and that the data is being received properly.

Here's a summary of the steps in the connection-oriented session—that three-way handshake—pictured in Figure 1.18:

- The first "connection agreement" segment is a request for *synchronization (SYN)*.
- The next segments *acknowledge (ACK)* the request and establish connection parameters—the rules—between hosts. These segments request that the receiver's sequencing is synchronized here as well so that a bidirectional connection can be formed.
- The final segment is also an acknowledgment, which notifies the destination host that the connection agreement has been accepted and that the actual connection has been established. Data transfer can now begin.

Flow Control

Since floods and losing data can both be tragic, we have a fail-safe solution in place known as *flow control*. Its job is to ensure data integrity at the Transport layer by allowing applications to request reliable data transport between systems. Flow control prevents a sending host on one side of the connection from overflowing the buffers in the receiving host. Reliable data

transport employs a connection-oriented communications session between systems, and the protocols involved ensure that the following will be achieved:

- The segments delivered are acknowledged back to the sender upon their reception.
- Any segments not acknowledged are retransmitted.
- Segments are sequenced back into their proper order upon arrival at their destination.
- A manageable data flow is maintained in order to avoid congestion, overloading, or worse, data loss.

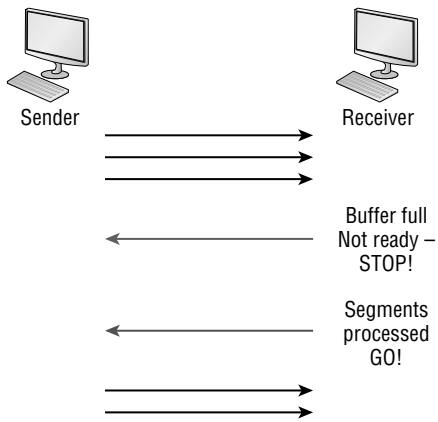
Because of the transport function, network flood control systems really work well. Instead of dumping and losing data, the Transport layer can issue a "not ready" indicator to the sender, or potential source of the flood. This mechanism works kind of like a stoplight, signaling the sending device to stop transmitting segment traffic to its overwhelmed peer. After the peer receiver processes the segments already in its memory reservoir—its buffer—it sends out a "ready" transport indicator. When the machine waiting to transmit the rest of its datagrams receives this "go" indicator, it resumes its transmission. The process is pictured in Figure 1.19.

In a reliable, connection-oriented data transfer, datagrams are delivered to the receiving host hopefully in the same sequence they're transmitted. A failure will occur if any data segments are lost, duplicated, or damaged along the way—a problem solved by having the receiving host acknowledge that it has received each and every data segment.

A service is considered connection-oriented if it has the following characteristics:

- A virtual circuit, or "three-way handshake" is set up.
- It uses sequencing.
- It uses acknowledgments.
- It uses flow control.

FIGURE 1.19 Transmitting segments with flow control



Windowing

Ideally, data throughput happens quickly and efficiently. And as you can imagine, it would be painfully slow if the transmitting machine had to actually wait for an acknowledgment after sending each and every segment! The quantity of data segments, measured in bytes, that the transmitting machine is allowed to send without receiving an acknowledgment is called a *window*.

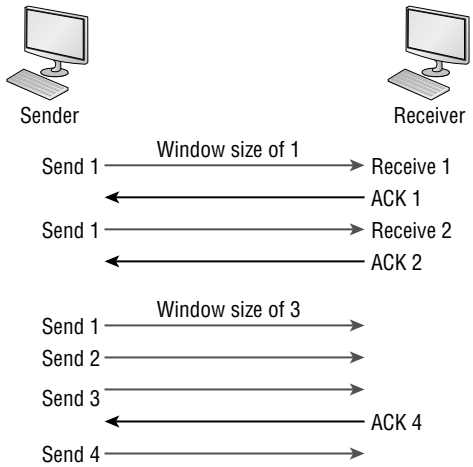
The size of the window controls how much information is transferred from one end to the other before an acknowledgment is required. While some protocols quantify information depending on the number of packets, TCP/IP measures it by counting the number of bytes.

As you can see in Figure 1.20, there are two window sizes—one set to 1 and one set to 3.

If you've configured a window size of 1, the sending machine will wait for an acknowledgment for each data segment it transmits before transmitting another one but will allow three to be transmitted before receiving an acknowledgment if the window size is set to 3.

In this simplified example, both the sending and receiving machines are workstations. Remember that in reality, the transmission isn't based on simple numbers but in the amount of bytes that can be sent!

FIGURE 1.20 Windowing



Acknowledgments

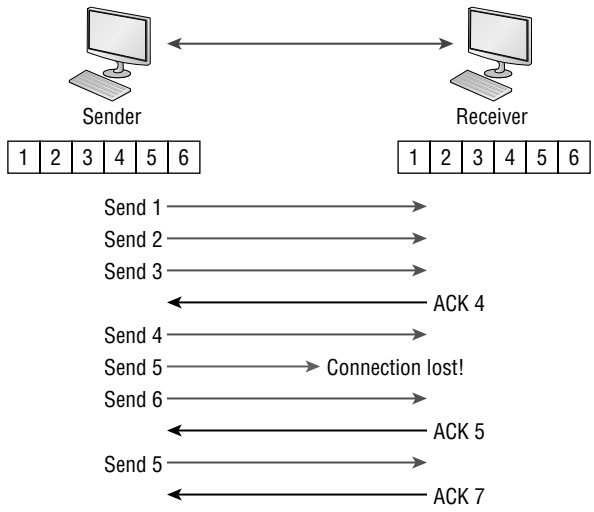
Reliable data delivery ensures the integrity of a stream of data sent from one machine to the other through a fully functional data link. It guarantees that the data won't be duplicated or lost. This is achieved through something called *positive acknowledgment with retransmission*—a technique that requires a receiving machine to communicate with the transmitting source by sending an acknowledgment message to the sender when it receives data. The sender documents each segment measured in bytes and then sends and

waits for this acknowledgment before sending the next segment. Also important is that when it sends a segment, the transmitting machine starts a timer and will retransmit if it expires before it gets an acknowledgment from the receiving end. Figure 1.21 shows the process I just described.

In the figure, the sending machine transmits segments 1, 2, and 3. The receiving node acknowledges that it has received them by requesting segment 4 (what it is expecting next). When it receives the acknowledgment, the sender then transmits segments 4, 5, and 6. If segment 5 doesn't make it to the destination, the receiving node acknowledges that event with a request for the segment to be resent. The sending machine will then resend the lost segment and wait for an acknowledgment, which it must receive in order to move on to the transmission of segment 7.

The Transport layer, working in tandem with the Session layer, also separates the data from different applications, an activity known as *session multiplexing*, and it happens when a client connects to a server with multiple browser sessions open. This is exactly what's taking place when you go someplace online like Amazon and click multiple links, opening them simultaneously to get information when comparison shopping. The client data from each browser session must be separate when the server application receives it, which is pretty slick technologically speaking, and it's the Transport layer to the rescue for that juggling act!

FIGURE 1.21 Transport layer reliable delivery



The Network Layer

The *Network layer*, or layer 3, manages device addressing, tracks the location of devices on the network, and determines the best way to move data. This means that it's up to the Network layer to transport traffic between devices that aren't locally attached. Routers,

which are layer 3 devices, are specified at this layer and provide the routing services within an internetwork.

Here's how that works: First, when a packet is received on a router interface, the destination IP address is checked. If the packet isn't destined for that particular router, it will look up the destination network address in the routing table. Once the router chooses an exit interface, the packet will be sent to that interface to be framed and sent out on the local network. If the router can't find an entry for the packet's destination network in the routing table, the router drops the packet.

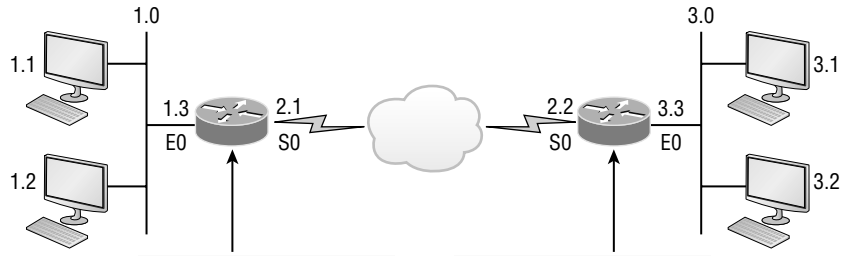
Data and route update packets are the two types of packets used at the Network layer:

Data packets These are used to transport user data through the internetwork. Protocols used to support data traffic are called routed protocols, and IPv4 and IPv6 are key examples.

Route update packets These packets are used to update neighboring routers about the networks connected to all routers within the internetwork. Protocols that send route update packets are called routing protocols; the most critical ones for CCNA are RIP, RIPv2, EIGRP, and OSPF. Route update packets are used to help build and maintain routing tables.

Figure 1.22 shows an example of a routing table. The routing table each router keeps and refers to includes the following information.

FIGURE 1.22 Routing table used in a router



Routing table		
NET	INT	Metric
1	E0	0
2	S0	0
3	S0	1

Routing table		
NET	INT	Metric
1	S0	1
2	S0	0
3	E0	0

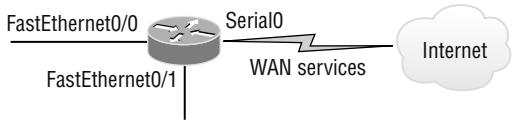
Network addresses Protocol-specific network addresses. A router must maintain a routing table for individual routing protocols because each routed protocol keeps track of a network with a different addressing scheme. For example, the routing tables for IPv4, IPv6, and IPX are completely different, so the router keeps a table for each one. Think of it as a street sign in each of the different languages spoken by the American, Spanish, and French people living on a street; the street sign would read Cat/Gato/Chat.

Interface The exit interface a packet will take when destined for a specific network.

Metric The distance to the remote network. Different routing protocols use different ways of computing this distance. Routing protocols like the Routing Information Protocol, or RIP, use hop count, which refers to the number of routers a packet passes through en route to a remote network. Others use bandwidth, delay of the line, or even tick count (1/18 of a second) to determine the best path for data to get to a given destination.

And as I mentioned earlier, routers break up broadcast domains, which means that by default, broadcasts aren't forwarded through a router. Do you remember why this is a good thing? Routers also break up collision domains, but you can also do that using layer 2, Data Link layer, switches. Because each interface in a router represents a separate network, it must be assigned unique network identification numbers, and each host on the network connected to that router must use the same network number. Figure 1.23 shows how a router works in an internetwork. Note that in the figure, each router LAN interface is a broadcast domain. Routers break up broadcast domains by default and provide WAN services.

FIGURE 1.23 A router in an internetwork



Here are some router characteristics that you should never forget:

- Routers, by default, will not forward any broadcast or multicast packets.
- Routers use the logical address in a Network layer header to determine the next-hop router to forward the packet to.
- Routers can use access lists, created by an administrator, to control security based on the types of packets allowed to enter or exit an interface.
- Routers can provide layer 2 bridging functions if needed and can simultaneously route through the same interface.
- Layer 3 devices—in this case, routers—provide connections between *virtual LANs (VLANs)*.
- Routers can provide *quality of service (QoS)* for specific types of network traffic.

The Data Link Layer

The *Data Link layer* provides for the physical transmission of data and handles error notification, network topology, and flow control. This means that the Data Link layer will ensure that messages are delivered to the proper device on a LAN using hardware addresses and will translate messages from the Network layer into bits for the Physical layer to transmit.

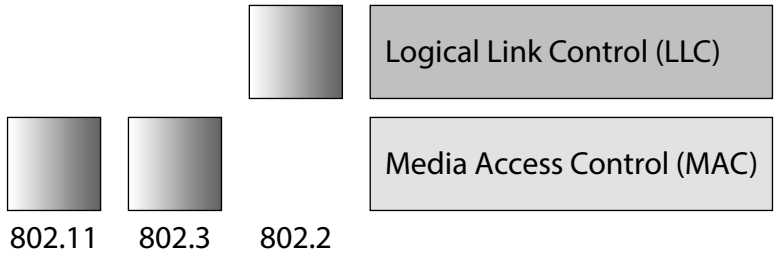
The Data Link layer formats the messages, each called a *data frame*, and adds a customized header containing the hardware destination and source address. This added information forms

a sort of capsule that surrounds the original message in much the same way that engines, navigational devices, and other tools were attached to the lunar modules of the Apollo project. These various pieces of equipment were useful only during certain stages of space flight and were stripped off the module and discarded when their designated stage was completed. The process of data traveling through networks is similar. I like this analogy!

Figure 1.24 shows the Data Link layer with the Ethernet and IEEE specifications. When you check it out, notice that the IEEE 802.2 standard is used in conjunction with and adds functionality to the other IEEE standards.

It's important for you to understand that routers, which work at the Network layer, don't care at all about where a particular host is located. They're only concerned about where networks are located and the best way to reach them—including remote ones. Routers are totally obsessive when it comes to networks, which in this case is a good thing! It's the Data Link layer that's responsible for the actual unique identification of each device that resides on a local network.

FIGURE 1.24 Data Link layer



For a host to send packets to individual hosts on a local network as well as transmit packets between routers, the Data Link layer uses hardware addressing. Each time a packet is sent between routers, it's framed with control information at the Data Link layer, but that information is stripped off at the receiving router and only the original packet is left completely intact. This framing of the packet continues for each hop until the packet is finally delivered to the correct receiving host. It's really important to understand that the packet itself is never altered along the route; it's only encapsulated with the type of control information required for it to be properly passed on to the different media types.

The IEEE Ethernet Data Link layer has two sub-layers:

Media Access Control (MAC) This first sub-layer is just above the Physical layer and defines how packets are placed on the media. Contention media access is "first come, first served" access where everyone shares the same bandwidth—hence the name. Physical addressing is defined here as well as logical topologies. What's a logical topology? It's the signal path through a physical topology. Line discipline, error notification (but not correction), the ordered delivery of frames, and optional flow control can also be used at this sub-layer.

Logical Link Control (LLC) The second sub-layer sits between the MAC sub-layer and the Network layer and is responsible for identifying Network layer protocols and then encapsulating them. An LLC header tells the Data Link layer what to do with a packet once a frame is received. It works like this: a host receives a frame and looks in the LLC header to find out where the packet is destined—for instance, the IP protocol at the Network layer. The LLC can also provide flow control and sequencing of control bits.

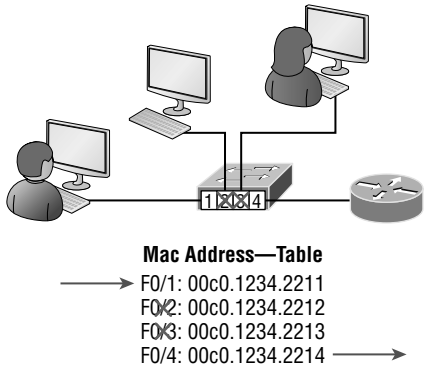
The switches and bridges I talked about near the beginning of the chapter both work at the Data Link layer and filter the network using hardware (MAC) addresses. I'll talk about these next.

Switches and Bridges at the Data Link Layer

Layer 2 switching is considered hardware-based bridging because it uses specialized hardware called an *application-specific integrated circuit (ASIC)*. ASICs can run up to high gigabit speeds with very low latency rates.

Bridges and switches read each frame as it passes through the network. The layer 2 device then puts the source hardware address in a filter table and keeps track of which port the frame was received on. This information (logged in the bridge's or switch's filter table) is what helps the machine determine the location of the specific sending device. Figure 1.25 shows a switch in an internetwork and how John is sending packets to the Internet and Sally doesn't hear his frames because she is in a different collision domain. The destination frame goes directly to the default gateway router, and Sally doesn't see John's traffic, much to her relief.

FIGURE 1.25 A switch in an internetwork



The real estate business is all about location, location, location, and it's the same way for both layer 2 and layer 3 devices. Though both need to be able to negotiate the network, it's crucial to remember that they're concerned with very different parts of it. Primarily, layer 3 machines (such as routers) need to locate specific networks, whereas layer 2 machines (switches and bridges) need to eventually locate specific devices. So, networks are to routers what individual devices are to switches and bridges. And routing tables that "map" the internetwork are for routers, just as filter tables that "map" individual devices are for switches and bridges.

After a filter table is built on the layer 2 device, it will forward frames only to the segment where the destination hardware address is located. If the destination device is on the same segment as the frame, the layer 2 device will block the frame from going to any other segments. If the destination is on a different segment, the frame can be transmitted only to that segment. This is called *transparent bridging*.

When a switch interface receives a frame with a destination hardware address that isn't found in the device's filter table, it will forward the frame to all connected segments. If the unknown device that was sent the "mystery frame" replies to this forwarding action, the switch updates its filter table regarding that device's location. But in the event the destination address of the transmitting frame is a broadcast address, the switch will forward all broadcasts to every connected segment by default.

All devices that the broadcast is forwarded to are considered to be in the same broadcast domain. This can be a problem because layer 2 devices propagate layer 2 broadcast storms that can seriously choke performance, and the only way to stop a broadcast storm from propagating through an internetwork is with a layer 3 device—a router!

The biggest benefit of using switches instead of hubs in your internetwork is that each switch port is actually its own collision domain. Remember that a hub creates one large collision domain, which is not a good thing! But even armed with a switch, you still don't get to just break up broadcast domains by default because neither switches nor bridges will do that. They'll simply forward all broadcasts instead.

Another benefit of LAN switching over hub-centered implementations is that each device on every segment plugged into a switch can transmit simultaneously. Well, at least they can as long as there's only one host on each port and there isn't a hub plugged into a switch port! As you might have guessed, this is because hubs allow only one device per network segment to communicate at a time.

The Physical Layer

Finally arriving at the bottom, we find that the *Physical layer* does two things: it sends bits and receives bits. Bits come only in values of 1 or 0—a Morse code with numerical values. The Physical layer communicates directly with the various types of actual communication media. Different kinds of media represent these bit values in different ways. Some use audio tones, while others employ *state transitions*—changes in voltage from high to low and low to high. Specific protocols are needed for each type of media to describe the proper bit patterns to be used, how data is encoded into media signals, and the various qualities of the physical media's attachment interface.

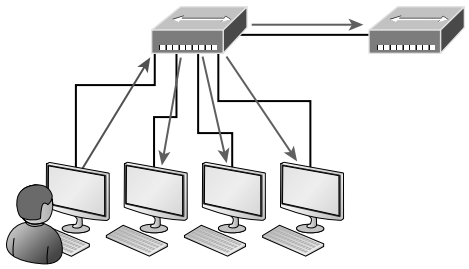
The Physical layer specifies the electrical, mechanical, procedural, and functional requirements for activating, maintaining, and deactivating a physical link between end systems. This layer is also where you identify the interface between the *data terminal equipment (DTE)* and the *data communication equipment (DCE)*. (Some old phone-company employees still call DCE "data circuit-terminating equipment.") The DCE is usually located at the service provider, while the DTE is the attached device. The services available to the DTE are most often accessed via a modem or *channel service unit/data service unit (CSU/DSU)*.

The Physical layer's connectors and different physical topologies are defined by the OSI as standards, allowing disparate systems to communicate. The Cisco exam objectives are interested only in the IEEE Ethernet standards.

Hubs at the Physical Layer

A hub is really a multiple-port repeater. A repeater receives a digital signal, reamplifies or regenerates that signal, and then forwards the signal out the other port without looking at any data. A hub does the same thing across all active ports: any digital signal received from a segment on a hub port is regenerated or reamplified and transmitted out all other ports on the hub. This means all devices plugged into a hub are in the same collision domain as well as in the same broadcast domain. Figure 1.26 shows a hub in a network and how when one host transmits, all other hosts must stop and listen.

FIGURE 1.26 A hub in a network



Hubs, like repeaters, don't examine any of the traffic as it enters or before it's transmitted out to the other parts of the physical media. And every device connected to the hub, or hubs, must listen if a device transmits. A physical star network, where the hub is a central device and cables extend in all directions out from it, is the type of topology a hub creates. Visually, the design really does resemble a star, whereas Ethernet networks run a logical bus topology, meaning that the signal has to run through the network from end to end.

Hubs and repeaters can be used to enlarge the area covered by a single LAN segment, but I really do not recommend going with this configuration! LAN switches are affordable for almost every situation and will make you much happier.

Exam Essentials

Define the OSI layers, understand the function of each, and describe how devices and networking protocols can be mapped to each layer. You must remember the seven layers of

the OSI model and what function each layer provides. The Application, Presentation, and Session layers are upper layers and are responsible for communicating from a user interface to an application. The Transport layer provides segmentation, sequencing, and virtual circuits. The Network layer provides logical network addressing and routing through an internetwork. The Data Link layer provides framing and placing of data on the network medium. The Physical layer is responsible for taking 1s and 0s and encoding them into a digital signal for transmission on the network segment.

Differentiate connection-oriented and connectionless network services and describe how each is handled during network communications. Connection-oriented services use acknowledgments and flow control to create a reliable session. More overhead is used than in a connectionless network service. Connectionless services are used to send data with no acknowledgments or flow control. This is considered unreliable.

Predict the Data Flow between Two Hosts across a Network

Once you create an internetwork by connecting your WANs and LANs to a router, you'll need to configure logical network addresses, like IP addresses, to all hosts on that internetwork for them to communicate successfully throughout it.

The term *routing* refers to taking a packet from one device and sending it through the network to another device on a different network. Routers don't really care about hosts—they only care about networks and the best path to each one of them. The logical network address of the destination host is key to get packets through a routed network. It's the hardware address of the host that's used to deliver the packet from a router and ensure that it arrives at the correct destination host.

Routing is irrelevant if your network has no routers because their job is to route traffic to all the networks in your internetwork, but rarely will your network have no routers! So here's an important list of the minimum factors a router must know to be able to effectively route packets:

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

The router learns about remote networks from neighboring routers or from an administrator. The router then builds a routing table, which is basically a map of the internetwork, and it describes how to find remote networks. If a network is directly connected, then the router already knows how to get to it.

But if a network isn't directly connected to the router, the router must use one of two ways to learn how to get to the remote network. The *static routing* method requires someone to hand-type all network locations into the routing table, which can be a pretty daunting task when used on all but the smallest of networks!

Conversely, when *dynamic routing* is used, a protocol on one router communicates with the same protocol running on neighboring routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If static routing is used, the administrator is responsible for updating all changes by hand onto all routers. Most people usually use a combination of dynamic and static routing to administer a large network.

Before we jump into the IP routing process, let's take a look at a very simple example that demonstrates how a router uses the routing table to route packets out of an interface. We'll be going into a more detailed study of the process soon, but I want to show you something called the "longest match rule" first. With it, IP will scan a routing table to find the longest match as compared to the destination address of a packet. Let's take a look at Figure 1.27 to get a picture of this process.

FIGURE 1.27 A simple routing example

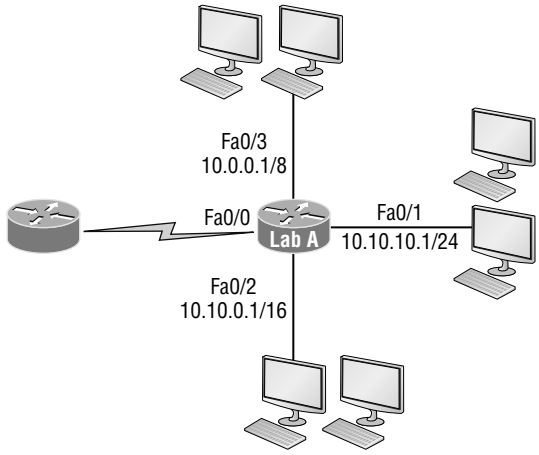


Figure 1.27 shows a simple network. Lab_A has four interfaces. Can you see which interface will be used to forward an IP datagram to a host with a destination IP address of 10.10.10.30?

By using the command `show ip route` on a router, we can see the routing table (map of the internetwork) that Lab A has used to make its forwarding decisions:

```
Lab_A#sh ip route
Codes: L - local, C - connected, S - static,
[output cut]
```

```
        10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
C       10.0.0.0/8 is directly connected, FastEthernet0/3
L       10.0.0.1/32 is directly connected, FastEthernet0/3
C       10.10.0.0/16 is directly connected, FastEthernet0/2
L       10.10.0.1/32 is directly connected, FastEthernet0/2
C       10.10.10.0/24 is directly connected, FastEthernet0/1
L       10.10.10.1/32 is directly connected, FastEthernet0/1
S*      0.0.0.0/0 is directly connected, FastEthernet0/0
```

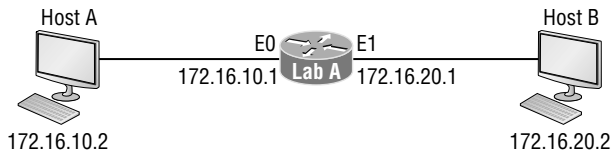
The C in the routing table output means that the networks listed are "directly connected," and until we add a routing protocol like RIPv2, OSPF, and so on, to the routers in our internetwork, or enter static routes, only directly connected networks will show up in our routing table. But wait—what about that L in the routing table—that's new, isn't it? Yes it is, because in the new Cisco IOS 15 code, Cisco defines a different route, called a local route. Each has a /32 prefix defining a route just for the one address. So in this example, the router has relied upon these routes that list their own local IP addresses to more efficiently forward packets to the router itself.

So let's get back to the original question: By looking at the figure and the output of the routing table, can you determine what IP will do with a received packet that has a destination IP address of 10.10.10.30? The answer is that the router will packet-switch the packet to interface FastEthernet 0/1, which will frame the packet and then send it out on the network segment. Based upon the longest match rule, IP would look for 10.10.10.30, and if that isn't found in the table, then IP would search for 10.10.10.0, then 10.10.0.0, and so on, until a route is discovered.

The IP Routing Process

The IP routing process is fairly simple and doesn't change, regardless of the size of your network. For a good example of this fact, I'll use Figure 1.28 to describe step-by-step what happens when Host A wants to communicate with Host B on a different network.

FIGURE 1.28 IP routing example using two hosts and one router



In Figure 1.28 a user on Host A pinged Host B's IP address. Routing doesn't get any simpler than this, but it still involves a lot of steps, so let's work through them now:

1. Internet Control Message Protocol (ICMP) creates an echo request payload, which is simply the alphabet in the data field.

2. ICMP hands that payload to Internet Protocol (IP), which then creates a packet. At a minimum, this packet contains an IP source address, an IP destination address, and a Protocol field with 01h. Don't forget that Cisco likes to use *0x* in front of hex characters, so this could also look like 0x01. This tells the receiving host to whom it should hand the payload when the destination is reached—in this example, ICMP.
3. Once the packet is created, IP determines whether the destination IP address is on the local network or a remote one.
4. Since IP has determined that this is a remote request, the packet must be sent to the default gateway so it can be routed to the remote network. The Registry in Windows is parsed to find the configured default gateway.
5. The default gateway of Host A is configured to 172.16.10.1. For this packet to be sent to the default gateway, the hardware address of the router's interface Ethernet 0, which is configured with the IP address of 172.16.10.1, must be known. Why? So the packet can be handed down to the Data Link layer, framed, and sent to the router's interface that's connected to the 172.16.10.0 network. Because hosts communicate only via hardware addresses on the local LAN, it's important to recognize that for Host A to communicate to Host B, it has to send packets to the Media Access Control (MAC) address of the default gateway on the local network.

MAC addresses are always local on the LAN and never go through and past a router.

6. Next, the Address Resolution Protocol (ARP) cache of the host is checked to see if the IP address of the default gateway has already been resolved to a hardware address.

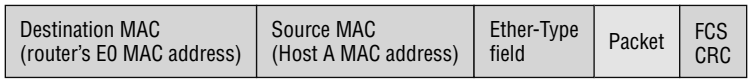
   If it has, the packet is then free to be handed to the Data Link layer for framing. Remember that the hardware destination address is also handed down with that packet. To view the ARP cache on your host, use the following command:

```
C:\>arp -a
Interface: 172.16.10.2 --- 0x3
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-05-06-31-b0     dynamic
```

   If the hardware address isn't already in the ARP cache of the host, an ARP broadcast will be sent out onto the local network to search for the 172.16.10.1 hardware address. The router then responds to the request and provides the hardware address of Ethernet 0, and the host caches this address.
7. Once the packet and destination hardware address are handed to the Data Link layer, the LAN driver is used to provide media access via the type of LAN being used, which is Ethernet in this case. A frame is then generated, encapsulating the packet with control information. Within that frame are the hardware destination and source addresses plus,

in this case, an Ether-Type field, which identifies the specific Network layer protocol that handed the packet to the Data Link layer. In this instance, it's IP. At the end of the frame is something called a Frame Check Sequence (FCS) field that houses the result of the cyclic redundancy check (CRC). The frame would look something like what I've detailed in Figure 1.29. It contains Host A's hardware (MAC) address and the destination hardware address of the default gateway. It does not include the remote host's MAC address—remember that!

FIGURE 1.29 Frame used from Host A to the Lab A router when Host B is pinged



8. Once the frame is completed, it's handed down to the Physical layer to be put on the physical medium (in this example, twisted-pair wire) one bit at a time.
9. Every device in the collision domain receives these bits and builds the frame. They each run a CRC and check the answer in the FCS field. If the answers don't match, the frame is discarded.
   - If the CRC matches, then the hardware destination address is checked to see if it matches (which, in this example, is the router's interface Ethernet 0).
   - If it's a match, then the Ether-Type field is checked to find the protocol used at the Network layer.
10. The packet is pulled from the frame, and what is left of the frame is discarded. The packet is handed to the protocol listed in the Ether-Type field—it's given to IP.
11. IP receives the packet and checks the IP destination address. Since the packet's destination address doesn't match any of the addresses configured on the receiving router itself, the router will look up the destination IP network address in its routing table.
12. The routing table must have an entry for the network 172.16.20.0 or the packet will be discarded immediately and an ICMP message will be sent back to the originating device with a destination network unreachable message.
13. If the router does find an entry for the destination network in its table, the packet is switched to the exit interface—in this example, interface Ethernet 1. The following output displays the Lab A router's routing table. The `C` means "directly connected." No routing protocols are needed in this network since all networks (all two of them) are directly connected.

```
Lab_A>sh ip route
C       172.16.10.0 is directly connected,    Ethernet0
L       172.16.10.1/32 is directly connected, Ethernet0
C       172.16.20.0 is directly connected,    Ethernet1
L       172.16.20.1/32 is directly connected, Ethernet1
```

14. The router packet-switches the packet to the Ethernet 1 buffer.
15. The Ethernet 1 buffer needs to know the hardware address of the destination host and first checks the ARP cache.
    - If the hardware address of Host B has already been resolved and is in the router's ARP cache, then the packet and the hardware address will be handed down to the Data Link layer to be framed. Let's take a look at the ARP cache on the Lab A router by using the `show ip arp` command:

```
Lab_A#sh ip arp
Protocol  Address     Age(min) Hardware Addr  Type   Interface
Internet  172.16.20.1   -     00d0.58ad.05f4  ARPA   Ethernet1
Internet  172.16.20.2   3     0030.9492.a5dd  ARPA   Ethernet1
Internet  172.16.10.1   -     00d0.58ad.06aa  ARPA   Ethernet0
Internet  172.16.10.2  12     0030.9492.a4ac  ARPA   Ethernet0
```

      The dash (-) signifies that this is the physical interface on the router. This output shows us that the router knows the 172.16.10.2 (Host A) and 172.16.20.2 (Host B) hardware addresses. Cisco routers will keep an entry in the ARP table for 4 hours.
    - Now if the hardware address hasn't already been resolved, the router will send an ARP request out E1 looking for the 172.16.20.2 hardware address. Host B responds with its hardware address, and the packet and destination hardware addresses are then both sent to the Data Link layer for framing.
16. The Data Link layer creates a frame with the destination and source hardware addresses, Ether-Type field, and FCS field at the end. The frame is then handed to the Physical layer to be sent out on the physical medium one bit at a time.
17. Host B receives the frame and immediately runs a CRC. If the result matches the information in the FCS field, the hardware destination address will be then checked next. If the host finds a match, the Ether-Type field is then checked to determine the protocol that the packet should be handed to at the Network layer—IP in this example.
18. At the Network layer, IP receives the packet and runs a CRC on the IP header. If that passes, IP then checks the destination address. Since a match has finally been made, the Protocol field is checked to find out to whom the payload should be given.
19. The payload is handed to ICMP, which understands that this is an echo request. ICMP responds to this by immediately discarding the packet and generating a new payload as an echo reply.
20. A packet is then created including the source and destination addresses, Protocol field, and payload. The destination device is now Host A.
21. IP then checks to see whether the destination IP address is a device on the local LAN or on a remote network. Since the destination device is on a remote network, the packet needs to be sent to the default gateway.

22. The default gateway IP address is found in the Registry of the Windows device, and the ARP cache is checked to see if the hardware address has already been resolved from an IP address.
23. Once the hardware address of the default gateway is found, the packet and destination hardware addresses are handed down to the Data Link layer for framing.
24. The Data Link layer frames the packet of information and includes the following in the header:
    - The destination and source hardware addresses
    - The Ether-Type field with 0x0800 (IP) in it
    - The FCS field with the CRC result in tow
25. The frame is now handed down to the Physical layer to be sent out over the network medium one bit at a time.
26. The router's Ethernet 1 interface receives the bits and builds a frame. The CRC is run, and the FCS field is checked to make sure the answers match.
27. Once the CRC is found to be okay, the hardware destination address is checked. Since the router's interface is a match, the packet is pulled from the frame and the Ether-Type field is checked to determine which protocol the packet should be delivered to at the Network layer.
28. The protocol is determined to be IP, so it gets the packet. IP runs a CRC check on the IP header first and then checks the destination IP address.

IP does not run a complete CRC as the Data Link layer does—it only checks the header for errors.

Since the IP destination address doesn't match any of the router's interfaces, the routing table is checked to see whether it has a route to 172.16.10.0. If it doesn't have a route over to the destination network, the packet will be discarded immediately. I want to take a minute to point out that this is exactly where the source of confusion begins for a lot of administrators because when a ping fails, most people think the packet never reached the destination host. But as we see here, that's not *always* the case. All it takes for this to happen is for even just one of the remote routers to lack a route back to the originating host's network and—*poof!*—the packet is dropped on the *return trip*, not on its way to the host!

Just a quick note to mention that when (and if) the packet is lost on the way back to the originating host, you will typically see a request timed-out message because it is an unknown error. If the error occurs because of a known issue, such as if a route is not in the routing table on the way to the destination device, you will see a destination unreachable message. This should help you determine if the problem occurred on the way to the destination or on the way back.

29. In this case, the router happens to know how to get to network 172.16.10.0—the exit interface is Ethernet 0—so the packet is switched to interface Ethernet 0.
30. The router then checks the ARP cache to determine whether the hardware address for 172.16.10.2 has already been resolved.
31. Since the hardware address to 172.16.10.2 is already cached from the originating trip to Host B, the hardware address and packet are then handed to the Data Link layer.
32. The Data Link layer builds a frame with the destination hardware address and source hardware address and then puts IP in the Ether-Type field. A CRC is run on the frame and the result is placed in the FCS field.
33. The frame is then handed to the Physical layer to be sent out onto the local network one bit at a time.
34. The destination host receives the frame, runs a CRC, checks the destination hardware address, and then looks into the Ether-Type field to find out to whom to hand the packet.
35. IP is the designated receiver, and after the packet is handed to IP at the Network layer, it checks the Protocol field for further direction. IP finds instructions to give the payload to ICMP, and ICMP determines the packet to be an ICMP echo reply.
36. ICMP acknowledges that it has received the reply by sending an exclamation point (!) to the user interface. ICMP then attempts to send four more echo requests to the destination host.

You've just experienced Todd's 36 easy steps to understanding IP routing. The key point here is that if you had a much larger network, the process would be the *same*. It's just that the larger the internetwork, the more hops the packet goes through before it finds the destination host.

It's super-important to remember that when Host A sends a packet to Host B, the destination hardware address used is the default gateway's Ethernet interface. Why? Because frames can't be placed on remote networks—only local networks. So packets destined for remote networks must go through the default gateway.

Let's take a look at Host A's ARP cache now:

```
C:\ >arp -a
Interface: 172.16.10.2 --- 0x3
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-05-06-31-b0     dynamic
  172.16.20.1           00-15-05-06-31-b0     dynamic
```

Did you notice that the hardware (MAC) address that Host A uses to get to Host B is the Lab A E0 interface? Hardware addresses are *always* local, and they never pass through a router's interface. Understanding this process is as important as air to you, so carve this into your memory!

The Cisco Router Internal Process

One more thing before we get to testing your understanding of my 36 steps of IP routing. I think it's important to explain how a router forwards packets internally. For IP to look up a

destination address in a routing table on a router, processing in the router must take place, and if there are tens of thousands of routes in that table, the amount of CPU time would be enormous. It results in a potentially overwhelming amount of overhead—think about a router at your ISP that has to calculate millions of packets per second and even subnets to find the correct exit interface! Even with the little network I'm using in this book, lots of processing would need to be done if there were actual hosts connected and sending data.

Cisco uses three types of packet-forwarding techniques.

Process switching This is actually how many people see routers to this day, because it's true that routers actually did perform this type of bare-bones packet switching back in 1990 when Cisco released its very first router. But those days when traffic demands were unimaginably light are long gone—not in today's networks! This process is now extremely complex and involves looking up every destination in the routing table and finding the exit interface for every packet. This is pretty much how I just explained the process in my 36 steps. But even though what I wrote was absolutely true in concept, the internal process requires much more than packet-switching technology today because of the millions of packets per second that must now be processed. So Cisco came up with some other technologies to help with the "big process problem."

Fast switching This solution was created to make the slow performance of process switching faster and more efficient. Fast switching uses a cache to store the most recently used destinations so that lookups are not required for every packet. It is important to know that this "cache" is information from already processed packets, meaning that fast switching must "process switch" a packet first. Nevertheless, when the exit interface of the destination device was cached, as well as the layer 2 header, performance was dramatically improved, but as our networks evolved with the need for even more speed, Cisco created yet another technology!

Cisco Express Forwarding (CEF) This is Cisco's newer creation, and it's the default packet-forwarding method used on all the latest Cisco routers. CEF proactively makes many different cache tables to help improve performance and is change triggered, not packet triggered. Translated, this means that when the network topology changes, the cache changes along with it.

Exam Essentials

Describe the basic IP routing process. You need to remember that the frame changes at each hop but that the packet is never changed or manipulated in any way until it reaches the destination device (the TTL field in the IP header is decremented for each hop, but that's it!).

List the information required by a router to successfully route packets. To be able to route packets, a router must know, at a minimum, the destination address, the location of neighboring routers through which it can reach remote networks, possible routes to all remote networks, the best route to each remote network, and how to maintain and verify routing information.

Identify the Appropriate Media, Cables, Ports, and Connectors to Connect Cisco Network Devices to Other Network Devices and Hosts in a LAN

The IEEE extended the 802.3 committee to three new committees known as 802.3u (Fast Ethernet), 802.3ab (Gigabit Ethernet on category 5), and then finally one more, 802.3ae (10 Gbps over fiber and coax). There are more standards evolving almost daily, such as the new 100 Gbps Ethernet (802.3ba)!

When designing your LAN, it's really important to understand the different types of Ethernet media available to you. Sure, it would be great to run Gigabit Ethernet to each desktop and 10 Gbps between switches, but you would need to figure out how to justify the cost of that network today! However, if you mix and match the different types of Ethernet media methods currently available, you can come up with a cost-effective network solution that works really great.

The *EIA/TIA* (Electronic Industries Alliance and the newer Telecommunications Industry Association) is the standards body that creates the Physical layer specifications for Ethernet. The EIA/TIA specifies that Ethernet use a *registered jack (RJ) connector* on *unshielded twisted-pair (UTP)* cabling (RJ45). But the industry is moving toward simply calling this an 8-pin modular connector.

Every Ethernet cable type that's specified by the EIA/TIA has inherent attenuation, which is defined as the loss of signal strength as it travels the length of a cable and is measured in decibels (dB). The cabling used in corporate and home markets is measured in categories. A higher-quality cable will have a higher-rated category and lower attenuation. For example, category 5 is better than category 3 because category 5 cables have more wire twists per foot and therefore less crosstalk. Crosstalk is the unwanted signal interference from adjacent pairs in the cable.

Here is a list of some of the most common IEEE Ethernet standards, starting with 10 Mbps Ethernet:

10Base-T (IEEE 802.3) 10 Mbps using category 3 unshielded twisted pair (UTP) wiring for runs up to 100 meters. Unlike with the 10Base-2 and 10Base-5 networks, each device must connect into a hub or switch, and you can have only one host per segment or wire. It uses an RJ45 connector (8-pin modular connector) with a physical star topology and a logical bus.

100Base-TX (IEEE 802.3u) 100Base-TX, most commonly known as Fast Ethernet, uses EIA/TIA category 5, 5E, or 6 UTP two-pair wiring. One user per segment; up to 100 meters long. It uses an RJ45 connector with a physical star topology and a logical bus.

100Base-FX (IEEE 802.3u) Uses fiber cabling 62.5/125-micron multimode fiber. Point-to-point topology; up to 412 meters long. It uses ST and SC connectors, which are media-interface connectors.

1000Base-CX (IEEE 802.3z) Copper twisted-pair, called twinax, is a balanced coaxial pair that can run only up to 25 meters and uses a special 9-pin connector known as the High Speed Serial Data Connector (HSSDC). This is used in Cisco's new Data Center technologies.

1000Base-T (IEEE 802.3ab) Category 5, four-pair UTP wiring up to 100 meters long and up to 1 Gbps.

1000Base-SX (IEEE 802.3z) The implementation of 1 Gigabit Ethernet running over multimode fiber-optic cable instead of copper twisted-pair cable, using short wavelength laser. Multimode fiber (MMF) using 62.5- and 50-micron core; uses an 850 nanometer (nm) laser and can go up to 220 meters with 62.5-micron, 550 meters with 50-micron.

1000Base-LX (IEEE 802.3z) Single-mode fiber that uses a 9-micron core and 1300 nm laser and can go from 3 kilometers up to 10 kilometers.

1000Base-ZX (Cisco standard) 1000BaseZX, or 1000Base-ZX, is a Cisco specified standard for Gigabit Ethernet communication. 1000BaseZX operates on ordinary single-mode fiber-optic links with spans up to 43.5 miles (70 km).

10GBase-T (802.3.an) 10GBase-T is a standard proposed by the IEEE 802.3an committee to provide 10 Gbps connections over conventional UTP cables, (category 5e, 6, or 7 cables). 10GBase-T allows the conventional RJ45 used for Ethernet LANs and can support signal transmission at the full 100-meter distance specified for LAN wiring.

Armed with the basics covered so far in this chapter, you're equipped to go to the next level and put Ethernet to work using various Ethernet cabling.

Ethernet Cabling

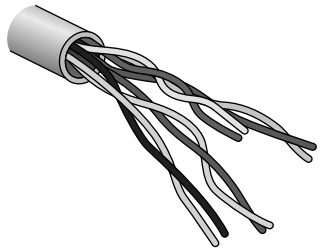
A discussion about Ethernet cabling is an important one, especially if you are planning on taking the Cisco exams. You need to really understand the following three types of cables:

- Straight-through cable
- Crossover cable
- Rolled cable

We will look at each in the following sections, but first, let's take a look at the most common Ethernet cable used today, the category 5 enhanced unshielded twisted pair (UTP), shown in Figure 1.30.

The category 5 Enhanced UTP cable can handle speeds up to a gigabit with a distance of up to 100 meters. Typically we'd use this cable for 100 Mbps and category 6 for a gigabit, but the category 5 Enhanced is rated for gigabit speeds and category 6 is rated for 10 Gbps!

FIGURE 1.30 Category 5 enhanced UTP cable



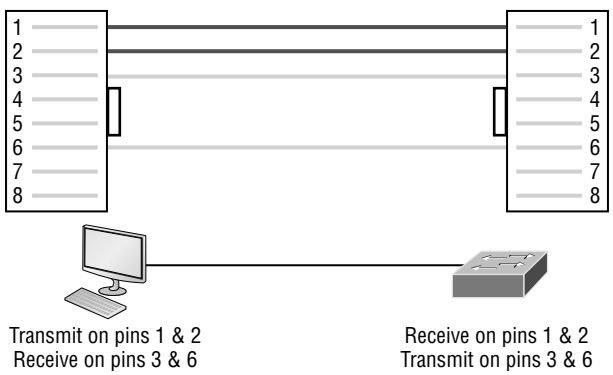
Straight-Through Cable

The *straight-through cable* is used to connect the following devices:

- Host to switch or hub
- Router to switch or hub

Four wires are used in straight-through cable to connect Ethernet devices. It's relatively simple to create this type, and Figure 1.31 shows the four wires used in a straight-through Ethernet cable.

FIGURE 1.31 Straight-through Ethernet cable



Notice that only pins 1, 2, 3, and 6 are used. Just connect 1 to 1, 2 to 2, 3 to 3, and 6 to 6 and you'll be up and networking in no time. However, remember that this would be a 10/100 Mbps Ethernet-only cable and wouldn't work with gigabit, voice, or other LAN or WAN technology.

Crossover Cable

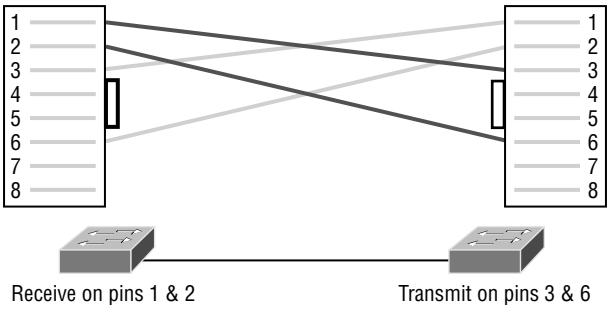
The *crossover cable* can be used to connect the following devices:

- Switch to switch
- Hub to hub

- Host to host
- Hub to switch
- Router direct to host
- Router to router

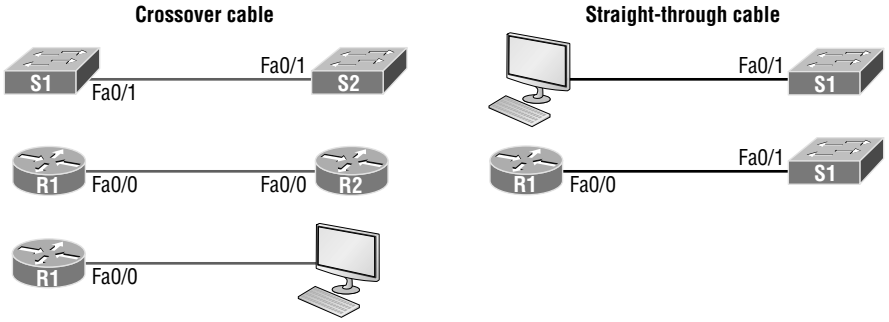
The same four wires used in the straight-through cable are used in this cable—we just connect different pins together. Figure 1.32 shows how the four wires are used in a crossover Ethernet cable.

FIGURE 1.32 Crossover Ethernet cable



Notice that instead of connecting 1 to 1, 2 to 2, and so on, here we connect pins 1 to 3 and 2 to 6 on each side of the cable. Figure 1.33 shows some typical uses of straight-through and crossover cables.

FIGURE 1.33 Typical uses for straight-through and crossover Ethernet cables



The crossover examples in Figure 1.33 are switch port to switch port, router Ethernet port to router Ethernet port, and PC Ethernet to router Ethernet port. For the straight-through examples I used PC Ethernet to switch port and router Ethernet port to switch port.

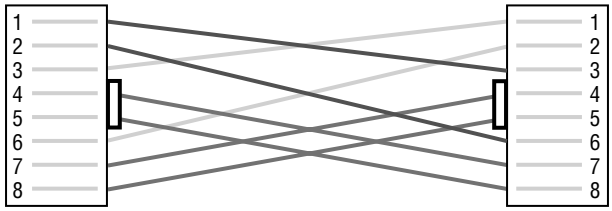
It's very possible to connect a straight-through cable between two switches, and it will start working because of autodetect mechanisms called auto-mdix. But be advised that the CCNA objectives do not typically consider autodetect mechanisms valid between devices!

UTP Gigabit Wiring (1000Base-T)

In 10Base-T and 100Base-T UTP wiring, only two wire pairs were used, but that is not good enough for Gigabit UTP transmission.

1000Base-T UTP wiring (Figure 1.34) requires four wire pairs and uses more advanced electronics so that each and every pair in the cable can transmit simultaneously. Even so, gigabit wiring is almost identical to my earlier 10/100 example, except that we'll use the other two pairs in the cable.

FIGURE 1.34 UTP Gigabit crossover Ethernet cable



For a straight-through cable, it's still 1 to 1, 2 to 2, and so on, up to pin 8. And in creating the gigabit crossover cable, you'd still cross 1 to 3 and 2 to 6, but you would add 4 to 7 and 5 to 8—pretty straightforward!

Rolled Cable

Although *rolled cable* isn't used to connect any Ethernet connections together, you can use a rolled Ethernet cable to connect a host EIA-TIA 232 interface to a router console serial communication (COM) port.

If you have a Cisco router or switch, you would use this cable to connect your PC, Mac, or a device like an iPad to the Cisco hardware. Eight wires are used in this cable to connect serial devices, although not all eight are used to send information, just as in Ethernet networking. Figure 1.35 shows the eight wires used in a rolled cable.

These are probably the easiest cables to make because you just cut the end off on one side of a straight-through cable, turn it over, and put it back on—with a new connector, of course!

Okay, once you have the correct cable connected from your PC to the Cisco router or switch console port, you can start your emulation program such as putty or SecureCRT to create a console connection and configure the device. Set the configuration as shown in Figure 1.36.

FIGURE 1.35 Rolled Ethernet cable

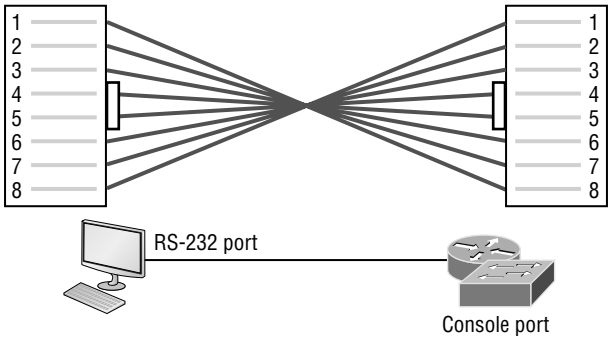
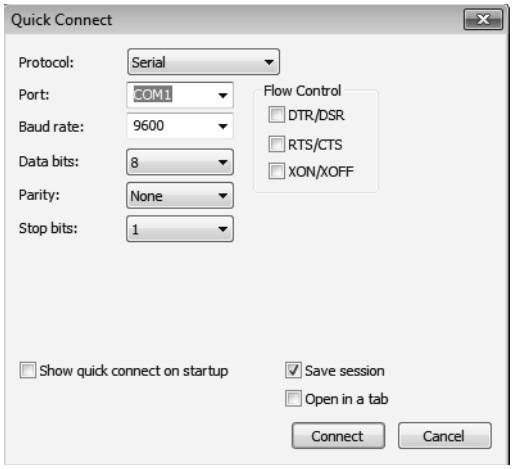


FIGURE 1.36 Configuring your console emulation program

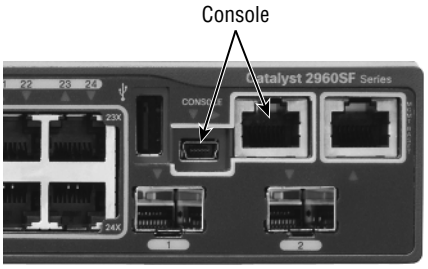


Notice that Bit Rate is set to 9600, Data Bits to 8, Parity to None, and Flow Control is set to None. At this point, you can click Connect and press the Enter key and you should be connected to your Cisco device console port.

Figure 1.37 shows a nice new 2960 switch with two console ports.

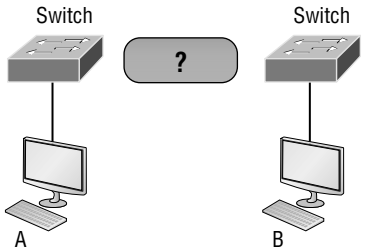
Notice that there are two console connections on this new switch—a typical original RJ45 connection and the newer mini type-B USB console. Remember that the new USB port supersedes the RJ45 port if you just happen to plug into both at the same time, and the USB port can have speeds up to 115,200 Kbps, which is awesome if you have to use Xmodem to update an IOS. I've even seen some cables that work on iPhones and iPads and allow them to connect to these mini USB ports!

FIGURE 1.37 Cisco 2960 console connections



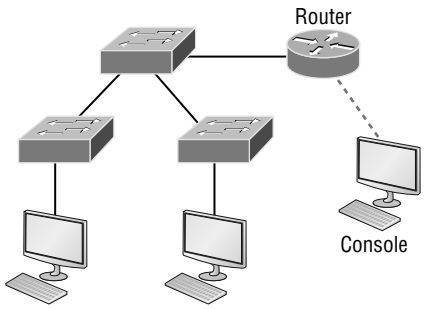
Now that you've seen the various RJ45 unshielded twisted-pair (UTP) cables, what type of cable is used between the switches in Figure 1.38?

FIGURE 1.38 RJ45 UTP cable question #1



In order for host A to ping host B, you need a crossover cable to connect the two switches together. But what types of cables are used in the network shown in Figure 1.39?

FIGURE 1.39 RJ45 UTP cable question #2



In Figure 1.39, there's a whole menu of cables in use. For the connection between the switches, we'd obviously use a crossover cable like we saw in Figure 1.34. The trouble is that you must understand that we have a console connection that uses a rolled cable. Plus, the connection from the router to the switch is a straight-through cable, as is true for the hosts to the switches. Keep in mind that if we had a serial connection, which we don't, we would use a V.35 to connect us to a WAN.

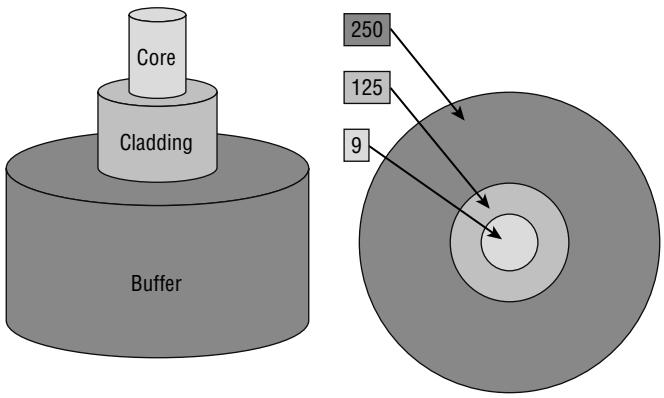
Fiber Optic

Fiber-optic cabling has been around for a long time and has some solid standards. The cable allows for very fast transmission of data, is made of glass (or even plastic!), is very thin, and works as a waveguide to transmit light between two ends of the fiber. Fiber optics has been used to go very long distances, as in intercontinental connections, but it is becoming more and more popular in Ethernet LAN networks due to the fast speeds available and because, unlike UTP, it's immune to interference like crosstalk.

Some main components of this cable are the core and the cladding. The core will hold the light and the cladding confines the light in the core. The tighter the cladding, the smaller the core, and when the core is small, less light will be sent but it can go faster and farther!

In Figure 1.40 you can see that there is a 9-micron core, which is very small and can be measured against a human hair, which is 50 microns.

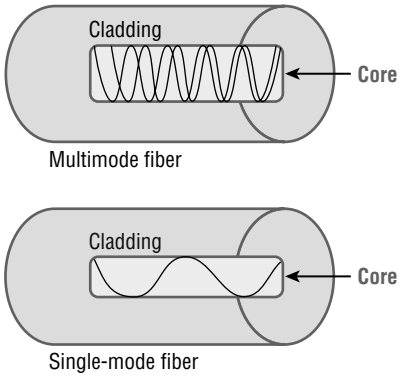
FIGURE 1.40 Typical fiber cable. Dimensions are in um (10^{-6} meters). Not to scale.



The cladding is 125 microns, which is actually a fiber standard that allows manufacturers to make connectors for all fiber cables. The last piece of this cable is the buffer, which is there to protect the delicate glass.

There are two major types of fiber optics: single-mode and multimode. Figure 1.41 shows the differences between multimode and single-mode fibers.

FIGURE 1.41 Multimode and single-mode fibers



Single-mode is more expensive, has a tighter cladding, and can go much farther distances than multimode. The difference comes in the tightness of the cladding, which makes a smaller core, meaning that only one mode of light will propagate down the fiber. Multimode is looser and has a larger core so it allows multiple light particles to travel down the glass. These particles have to be put back together at the receiving end, so distance is less than that with single-mode fiber, which allows only very few light particles to travel down the fiber.

There are about 70 different connectors for fiber, and Cisco uses a few different types. Looking back at Figure 1.37, the two bottom ports are referred to as small form-factor pluggables, or SFPs.

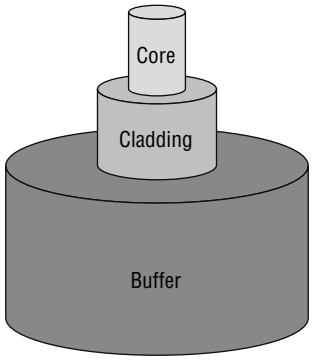
Exam Essentials

Identify the IEEE physical standards for Ethernet cabling. These standards describe the capabilities and physical characteristics of various cable types and include but are not limited to 10Base-2, 10Base-5, and 10Base-T.

Differentiate types of Ethernet cabling and identify their proper application. The three types of cables that can be created from an Ethernet cable are straight-through (to connect a PC's or router's Ethernet interface to a hub or switch), crossover (to connect hub to hub, hub to switch, switch to switch, or PC to PC), and rolled (for a console connection from a PC to a router or switch).

Review Questions

1. What cable type is shown in the following image?



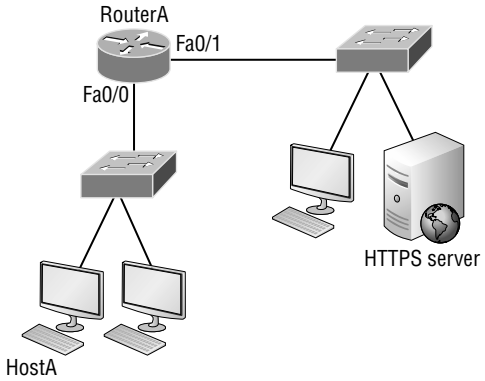
   A. Fiber optic
   B. Rollover
   C. Coaxial
   D. Full-duplex

2. Which of the following statements is/are true with regard to the device shown below?

   A. It includes one collision domain and one broadcast domain.
   B. It includes one collision domain and 10 broadcast domains.
   C. It includes 10 collision domains and one broadcast domain.
   D. It includes one collision domain and 10 broadcast domains.
   E. It includes 10 collision domains and 10 broadcast domains.

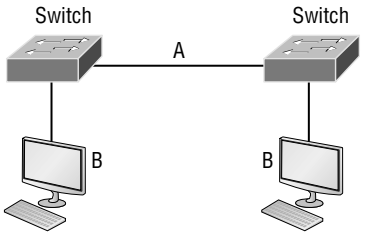
3. Which of the following Application layer protocols sets up a secure session that's similar to Telnet?
   A. FTP
   B. SSH
   C. DNS
   D. DHCP

4. What destination addresses will be used by HostA to send data to the HTTPS server as shown in the following network? (Choose two.)



   A. The IP address of the switch
   B. The MAC address of the remote switch
   C. The IP address of the HTTPS server
   D. The MAC address of the HTTPS server
   E. The IP address of RouterA's Fa0/0 interface
   F. The MAC address of RouterA's Fa0/0 interface

5. In the following diagram, identify the cable types required for connections A and B.

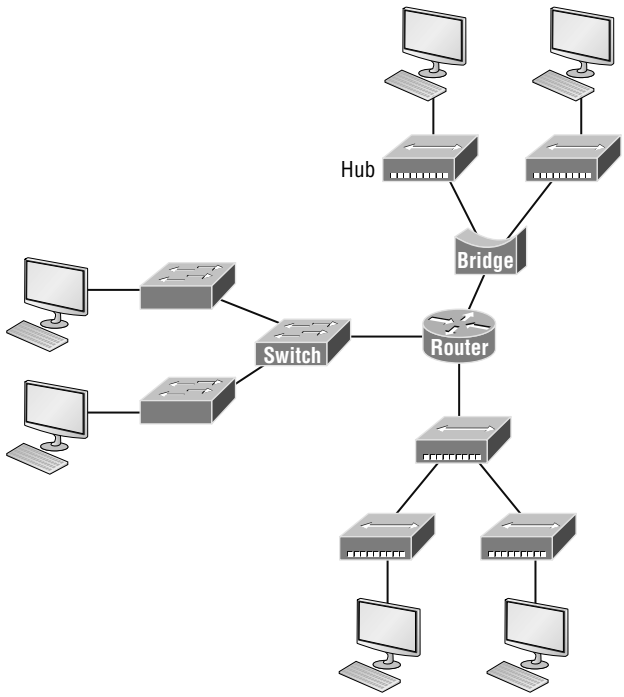


   A. A crossover, B crossover
   B. A crossover, B straight through
   C. A straight through, B straight through
   D. A straight through, B crossover

6. When a packet is routed across a network, the _____ in the frame changes at every hop while the _____ in the packet does not.
   - **A.** MAC address, IP address
   - **B.** IP address, MAC address
   - **C.** Port number, IP address
   - **D.** IP address, port number

7. What must happen if a DHCP IP conflict occurs?
   - **A.** Proxy ARP will fix the issue.
   - **B.** The client uses a gratuitous ARP to fix the issue.
   - **C.** The administrator must fix the conflict by hand at the DHCP server.
   - **D.** The DHCP server will reassign new IP addresses to both computers.

8. How many collision domains are present in the following diagram?



   - **A.** 8
   - **B.** 9
   - **C.** 10
   - **D.** 11

9. When a router looks up the destination in the routing table for every single packet it is called _____ .
   - **A.** dynamic switching
   - **B.** Fast switching
   - **C.** Process switching
   - **D.** Cisco Express Forwarding

10. What protocol is used to find the hardware address of a local device?
   - **A.** RARP
   - **B.** ARP
   - **C.** IP
   - **D.** ICMP
   - **E.** BootP