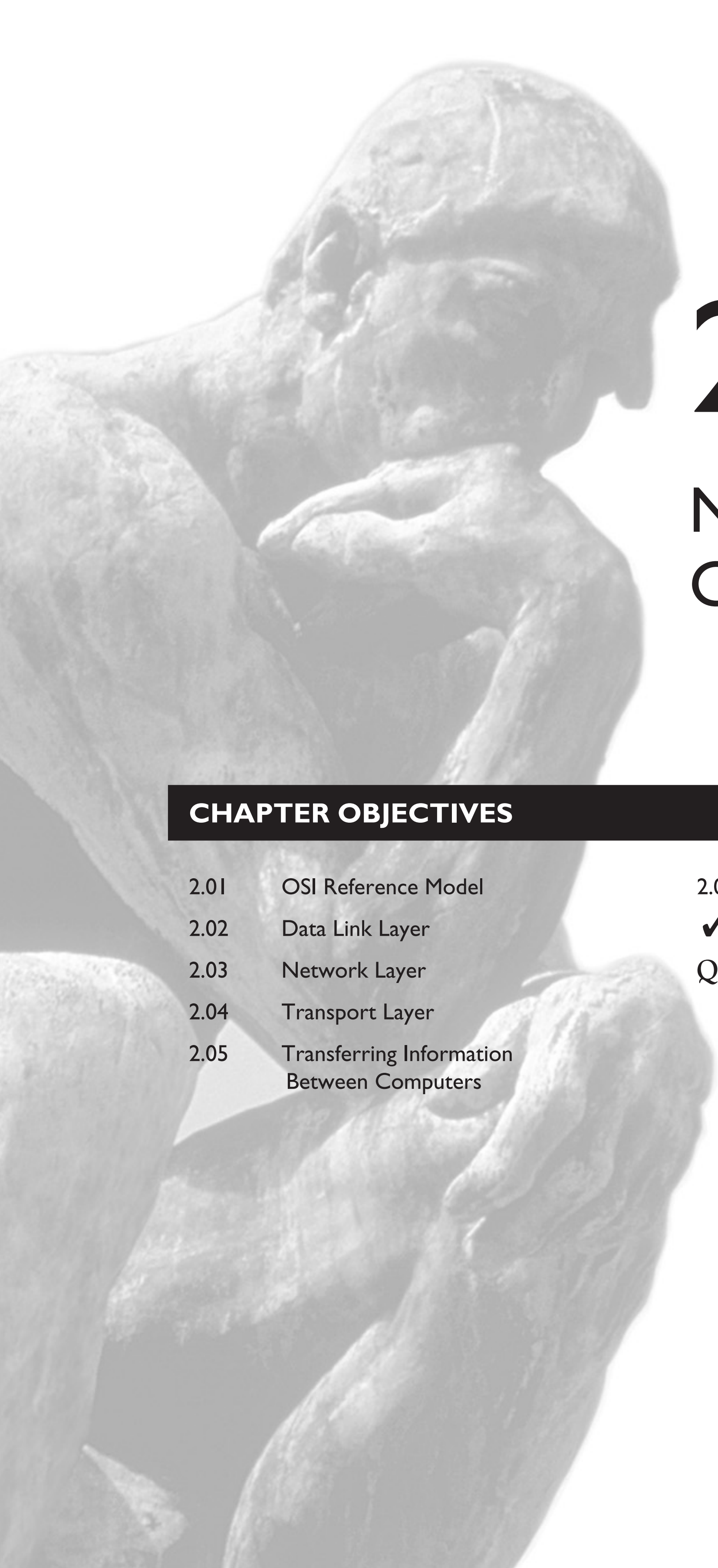


CCNA™

CISCO® CERTIFIED NETWORK ASSOCIATE

2

Networking Concepts

CHAPTER OBJECTIVES

Before considering how to configure Cisco routers and switches, you must be introduced to basic networking concepts you'll need to understand in order to grasp the advanced concepts discussed in later chapters. The OSI Reference Model is the best place to start, since it will help you understand how information is transferred between networking devices. Of the seven layers in the OSI Reference Model, be especially sure to understand how the bottom three layers function, since most networking devices function at these layers. This chapter discusses information flow, as well as Cisco's three-tiered hierarchical model, which is used to design scalable, flexible, and easy-to-troubleshoot-and-maintain networks.

CERTIFICATION OBJECTIVE 2.01

OSI Reference Model

The International Organization for Standardization (ISO) developed the Open Systems Interconnection (OSI) Reference Model to describe how information is transferred from one machine to another, from the point when a user enters information using a keyboard and mouse to when that information is converted to electrical or light signals transferred along a piece of wire or radio waves transferred through the air. It is important to understand that the OSI Reference Model describes concepts and terms in a general manner, and that many network protocols, such as IP and IPX, fail to fit nicely into the scheme explained in ISO's model. Therefore, the OSI Reference Model is most often used as a teaching and troubleshooting tool. By understanding the basics of the OSI Reference Model, you can apply these to real protocols to gain a better understanding of them as well as to more easily troubleshoot problems.

Advantages

ISO developed the seven-layer model to help vendors and network administrators gain a better understanding of how data is handled and transported between networking devices, as well as to provide a guideline for the implementation of new networking standards and technologies. To assist in this process, the OSI Reference Model breaks the network communication process into seven simple steps. It thus

- Defines the process for connecting two layers, promoting interoperability between vendors.

- Separates a complex function into simpler components.
- Allows vendors to compartmentalize their design efforts to fit a modular design, which eases implementations and simplifies troubleshooting.

A PC is a good example of a modular device. For instance, a PC typically contains the following components: case, motherboard with processor, monitor, keyboard, mouse, disk drive, CD-ROM drive, floppy drive, RAM, video card, Ethernet card, etc. If one component breaks, it is very easy to figure out which component failed and replace the single component. This simplifies your troubleshooting process. Likewise, when a new CD-ROM drive becomes available, you don't have to throw away the current computer to use the new device— you just need to cable it up and add a software driver to your operating system to interface with it. The OSI Reference Model builds upon these premises.

Layer Definitions

There are seven layers in the OSI Reference Model, shown in Figure 2-1: application, presentation, session, transport, network, data link, and physical. The functions of the application, presentation, and session layers are typically part of the user's application. The transport, network, data link, and physical layers are responsible for moving information back and forth between these higher layers.

Each layer is responsible for a specific process or role. Remember that the seven layers are there to help you understand the transformation process that data will

FIGURE 2-1

OSI Reference Model

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical

undergo as it is transported to a remote networking device. Not every networking protocol will fit exactly into this model. For example, TCP/IP has four layers. Some layers are combined into a single layer; for instance, TCP/IP's application layer contains the functionality of the OSI Reference Model's application, presentation, and session layers. The following sections go into more detail concerning the seven layers of the OSI Reference Model.

Application Layer

The seventh layer, or topmost layer, of the OSI Reference Model is the *application* layer. It provides the interface that a person uses to interact with the application. This interface can be command-line-based or graphics-based. Cisco IOS routers and switches have a command-line interface (CLI), whereas a web browser uses a graphical interface.

exam watch

The top (seventh) layer of the OSI Reference Model is the application layer. It provides the user interface. Examples of TCP/IP applications include telnet, FTP, HTTP, and SMTP.

Note that in the OSI Reference Model, the application layer refers to applications that are network-aware. There are thousands of computer applications, but not all of these can transmit information across a network. This situation is changing rapidly, however. Five years ago, there was a distinct line between applications that could and couldn't perform network functions. A good example of this was word processing programs, like Microsoft Word—they were built to perform one process: word processing. Today, however, many applications—Microsoft Word, for instance—have embedded objects that don't necessarily have to be on the same computer. There are many, many examples of application layer programs. The most common are telnet, FTP, web browsers, and e-mail.

Presentation Layer

The sixth layer of the OSI Reference Model is the *presentation* layer. The presentation layer is responsible for defining how information is presented to the user in the interface that they are using. This layer defines how various forms of text, graphics, video, and/or audio information are presented to the user. For example, text is represented in two different forms: ASCII and EBCDIC. ASCII (the American Standard Code for Information Interchange, used by most devices today) uses seven bits to represent characters. EBCDIC (Extended Binary-Coded Decimal Interchange Code, developed by IBM) is still used in mainframe environments to represent characters. Text can also be shaped by different elements, such as font, underline, italic, and bold.

exam watch

The presentation layer determines how data is represented to the user. Examples of presentation layer protocols and standards include ASCII, BMP, GIF, JPEG, WAV, AVI, and MPEG.

There are different standards for representing graphical information—BMP, GIF, JPEG, TIFF, and others. This variety of standards is also true of audio (WAV and MIDI) and video (WMV, AVI, and MPEG). There are literally hundreds of standards for representing information that a user sees in their application. Probably one of the best examples of applications that have a very clear presentation function is a web browser, since it has many special marking codes that define how data should be represented to the user.

The presentation layer can also provide encryption to secure data from the application layer; however, this it not common with today's methods of security, since this type of encryption is performed in software and requires a lot of CPU cycles to perform.

Session Layer

The fifth layer of the OSI Reference Model is the *session* layer. The session layer is responsible for initiating the setup and teardown of connections. In order to perform these functions, the session layer must determine whether data stays local to a computer or must be obtained or sent to a remote networking device. In the latter case, the session layer initiates the connection. The session layer is also responsible for differentiating among multiple network connections, ensuring that data is sent across the correct connection as well as taking data from a connection and forwarding it to the correct application.

exam watch

The session layer is responsible for setting up and tearing down network connections. Examples include RPCs and NFS.

The actual mechanics of this process, however, are implemented at the transport layer. To set up connections or tear down connections, the session layer communicates with the transport layer. Remote Procedure Call (RPC) is an example of an IP session protocol; the Network File System (NFS), which uses RPC, is an example application at this layer.

Transport Layer

The fourth layer of the OSI Reference Model is the *transport* layer. The transport layer is responsible for the actual mechanics of a connection, where it can provide both *reliable* and *unreliable* delivery of data. For reliable connections, the transport layer is responsible for error detection and correction: when an error is detected, the transport layer will resend the data, thus providing the correction. For unreliable connections, the transport layer provides only error detection—error correction is left up to one of the higher layers (typically the application layer). In this sense, unreliable connections attempt to provide a best-effort delivery—if the data makes it there, that's great, and if it doesn't, oh well!

exam watch

The fourth layer, the transport layer, provides both guaranteed data delivery and no guarantee of data delivery. Examples include IP's TCP and UDP protocols.

Examples of a reliable transport protocol are TCP/IP's Transmission Control Protocol (TCP) and IPX's SPX (Sequenced Packet Exchange) protocol. TCP/IP's User Datagram Protocol (UDP) is an example of a protocol that uses unreliable connections. Actually, IPX and IP themselves are examples of protocols that provide unreliable connections, even though they operate at the network, and not transport, layer. In IPX's case, if a reliable connection is needed, SPX is used. For IP, if a reliable connection is needed, TCP is used at the transport layer. The transport layer together with its mechanics is discussed in more depth in the section "Transport Layer" later in this chapter.

Network Layer

The third layer of the OSI Reference Model is the network layer. The network layer provides quite a few functions. First, it provides for a logical topology of your network using logical, or layer-3, addresses. These addresses are used to group machines together. As you will see in Chapter 3, these addresses have two components: a network component and a host component. The network component is used to group devices together. Layer-3 addresses allow devices that are on the same or different media types to communicate with each other. Media types define types of connections, such as Ethernet, Token Ring, or serial. These are discussed in the section "Data Link Layer" later in this chapter.

To move information between devices that have different network numbers, a *router* is used. Routers use information in the logical address to make intelligent decisions about how to reach a destination. Routing is discussed in more depth in Chapters 9, 10, and 11.

exam watch

The network layer provides a logical topology and layer-3 addresses. Routers function at the network layer. Layer-3 protocols include TCP/IP, IPX, and AppleTalk.

Examples of network layer protocols include AppleTalk, DECnet, IPX, TCP/IP (or IP, for short), Vines, and XNS. The network layer is discussed in much more depth in the section "Network Layer" later in this chapter.

Data Link Layer

The second layer in the OSI Reference Model is the *data link* layer. Whereas the network layer provides for logical addresses for devices, the data link layer provides for physical, or hardware, addresses. These hardware addresses are commonly called Media Access Control (MAC) addresses. The data link layer also defines how a networking device accesses the media that it is connected as well as defining the media's frame type. This includes the fields and components of the data link layer, or layer-2, frame. This communication is only for devices on the same data link layer media type (or same piece of wire). To traverse media types, Ethernet to Token Ring, for instance, typically a router is used.

The data link layer is also responsible for taking bits (binary 1's and 0's) from the physical layer and reassembling them into the original data link layer frame. The data link layer does error detection and will discard bad frames. It typically does not perform error correction, as TCP/IP's TCP protocol does; however, some data link layer protocols do support error correction functions.

Examples of data link layer protocols and standards for local area network (LAN) connections include IEEE's 802.2, 802.3, and 802.5; Ethernet II; and ANSI's FDDI. Examples of WAN connections include ATM, Frame Relay, HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol), SDLC (Synchronous Data Link Control), SLIP (Serial Line Internet Protocol), and X.25. Bridges, switches, and network interface controllers or cards (NICs) are the primary networking devices functioning at the data link layer, which is discussed in more depth in the section "Data Link Layer" later in this chapter.

exam watch

The data link layer defines hardware (MAC) addresses as well as the communication process that occurs within a media type. Switches and bridges function at the data link layer. Examples of data link layer protocols and standards include IEEE's 802.2, 802.3, Ethernet II, HDLC, PPP, and Frame Relay.

Physical Layer

The first, or bottommost, layer of the OSI Reference Model is the *physical* layer. The physical layer is responsible for the physical mechanics of a network connection, which include the following:

- The type of interface used on the networking device
- The type of cable used for connecting devices
- The connectors used on each end of the cable
- The pin-outs used for each of the connections on the cable

The type of interface is commonly called a NIC. A NIC can be a physical card that you put into a computer, like a 10BaseT Ethernet card, or a fixed interface on a switch, like a 100BaseTX port on a Cisco Catalyst 1900 series switch.

The physical layer is also responsible for how binary information is converted to a physical layer signal. For example, if the cable uses copper as a transport medium, the physical layer defines how binary 1's and 0's are converted into an electrical signal by using different voltage levels. If the cable uses fiber, the physical layer defines how 1's and 0's are represented using an LED or laser with different light frequencies.

Data communications equipment (DCE) terminates a physical WAN connection and provides clocking and synchronization of a connection between two locations and connects to a DTE. The DCE category includes equipment such as CSU/DSUs, NT1s, and modems. Data terminal equipment (DTE) is an end-user device, such as a router or a PC, that connects to the WAN via the DCE device. In some cases, the function of the DCE may be built into the DTE's physical interface. For instance, certain Cisco routers can be purchased with built-in NT1s or CSU/DSUs in their WAN interfaces. Normally, the terms DTE and DCE are used to describe WAN components, but they are sometimes used to describe LAN connections. For instance, in a LAN connection, a PC, file server, or router is sometimes referred to as a DTE, and a switch or bridge as a DCE.

exam watch

The physical layer defines physical properties for connections and communication, including wires (UTP and fiber) and connectors (RJ-45 and DB-9). A hub and a repeater are examples of devices that function at the physical layer. A repeater is used to physically extend a single segment, while a hub, which is also a repeater, connects many segments together.

Examples of physical layer standards include the following cable types: Category-3, -5, and -5E; EIA/TIA-232, -449, and -530; multimode and single-mode fiber (MMF and SMF); Type-1; and others. Interface connectors include the following: AUI, BNC, DB-9, DB-25, DB-60, RJ-11, RJ-45, and others. A hub and a repeater are examples of devices that function at the physical layer.

Fiber Cabling LANs typically use either copper or fiber-optic cabling. Copper cabling is discussed in more depth in the section "Ethernet" later in this chapter. Fiber-optic cabling uses light-emitting diodes (LEDs) and lasers to transmit data. With this transmission, light is used to represent binary 1's and 0's: if there is light on the wire, this represents a 1; if there is no light, this represents a 0.

Fiber-optic cabling is typically used to provide very high speeds and to span connections across very large distances. For example, speeds of 100Gbps and distances of over 10 kilometers are achievable through the use of fiber—copper cannot come close to these feats. However, fiber-optic cabling does have its disadvantages: it is expensive, difficult to troubleshoot, difficult to install, and less reliable than copper.

exam watch

Fiber cabling is not affected by electromagnetic interference (EMI), whereas copper cabling is.

Two types of fiber are used for connections: multimode and single-mode. Multimode fiber has a fiber thickness of either 850 or 1300 nanometers (nm), and the light signal is typically provided by an LED. When transmitting a signal, the light source is bounced off of the inner cladding (shielding) surrounding the fiber. Multimode fiber can achieve speeds in the hundreds of Mbps range, and many signals can be generated per fiber. Single-mode fiber has a fiber thickness of 1300 or 1550 nm and uses a laser as the light source. Because lasers provide a higher output than LEDs, single-mode fiber can span over 10 kilometers and have speeds up to 100Gbps. With single-mode fiber, only one signal is used per fiber.

The last few years have seen many advances in the use and deployment of fiber. One major enhancement is wave division multiplexing (WDM) and dense WDM (DWDM). WDM allows more than two wavelengths (signals) on the same piece of fiber, increasing the number of connections. DWDM allows yet more wavelengths, which are more closely spaced together: more than 200 wavelengths can be multiplexed into a light stream on a single piece of fiber.

Obviously, one of the advantages of DWDM is that it provides flexibility and transparency of the protocols and traffic carried across the fiber. For example, one wavelength can be used for a point-to-point connection, another for an Ethernet connection, another for an IP connection, and yet another for an ATM connection.

Use of DWDM provides scalability and allows carriers to provision new connections *without* having to install new fiber lines, so they can add new connections in a very short period when you order them.

Let's talk about some of the terms used in fiber and how they affect distance and speed. First, you have the cabling, which provides the protective outer coating as well as the inner cladding. The inner cladding is denser to allow the light source to bounce off of it. In the middle of the cable is the fiber itself, which is used to transmit the signal. The index of refraction (IOR) affects the speed of the light source: it's the ratio of the speed of light in a vacuum to the speed of light in the fiber. In a vacuum, there are no variables that affect the transmission; however, anytime you send something across a medium like fiber or copper, the media itself will exhibit properties that will affect the transmission, causing possible delays. IOR is used to measure these differences: basically, IOR measures the density of the fiber. The more dense the fiber is, the slower the light travels through the fiber.

The *loss factor* is used to describe any signal loss in the fiber before the light source gets to the end of the fiber. *Connector loss* is a loss that occurs when a connector joins two pieces of fibers: a slight signal loss is expected. Also, the longer the fiber, the greater the likelihood that the signal strength will have decreased when it reaches the end of the cable. This is called *attenuation*. Two other terms, microbending and macrobending, describe signal degradation.

Microbending is when a wrinkle in the fiber, typically where the cable is slightly bent, causes a distortion in the light source. *Macrobending* is when there is leakage of the light source from the fiber, typically from a bend in the fiber cable. To overcome this problem over long distances, *optical amplifiers* can be used. They are similar to an Ethernet repeater. A good amplifier, such as an erbium-doped fiber amplifier (EDFA), coverts a light source directly to another light source, providing for the best reproduction of the original signal. Other amplifiers convert light to an electrical signal and then back to light, which can cause a degradation in signal quality.

Two main standards are used to describe the transmission of signals across a fiber: SONET (Synchronous Optical Network) and SDH (Synchronous Digital Hierarchy). SONET is defined by the Exchange Carriers Standards Association (ECSA) and American National Standards Institute (ANSI) and is typically used in North America. SDH is an international standard used throughout most of the world (with the exception of North America). Both of these standards define the physical layer framing used to transmit light sources, which also includes overhead for the transmission. There are three types of overhead:

- **Section overhead (SOH)** Overhead for the link between two devices, such as repeaters

- **Line overhead (LOH)** Overhead for one or more sections connecting network devices, such as hubs
- **Path overhead (POH)** Overhead for one or more lines connecting two devices that assemble and disassemble frames, such as carrier switches or a router's fiber interface

Typically, either a ring or point-to-point topology is used to connect the devices. With carrier MAN networks, the most common implementation is through the use of rings. Autoprotection switching (APS) can be used to provide line redundancy: in case of failure on a primary line, a secondary line can automatically be utilized. Table 2-1 contains an overview of the more common connection types for SONET and SDH. Please note that SONET uses STS and that SDH uses STM to describe the signal.

Wireless Wireless transmission has been used for a very long time to transmit data by using infrared radiation, microwaves, or radio waves through a medium like air. With this type of connection, no wires are used. Typically, three terms are used to group different wireless technologies: narrowband, broadband, and circuit/packet data. Whenever you are choosing a wireless solution for your WAN or LAN, you should always consider the following criteria: speed, distance, and number of devices to connect.

Narrowband solutions typically require a license and operate at a low data rate. Only one frequency is used for transmission: 900 MHz, 2.4 GHz, or 5 GHz. Other technologies—household wireless phones, for instance—also use these technologies. Through the use of spread spectrum, higher data rates can be achieved by spreading the signal across multiple frequencies. However, transmission of these signals is typically limited to a small area, like a campus network.

TABLE 2-1 Fiber Connection Types

Common Term	SONET Term	SDH Term	Connection Rate
OC-1	STS-1	--	51.84 Mbps
OC-3	STS-3	STM-1	155.52 Mbps
OC-12	STS-12	STM-4	622.08 Mbps
OC-48	STS-48	STM-16	2,488.32 Mbps
OC-192	STS-192	STM-64	9,953.28 Mbps

The broadband solutions fall under the heading of the Personal Communications Service (PCS). They provide lower data rates than narrowband solutions, cost about the same, but provide broader coverage. With the right provider, you can obtain national coverage. Sprint PCS is an example of a carrier that provides this type of solution.

Circuit and packet data solutions are based on cellular technologies. They provide lower data rates than the other two and typically have higher fees for each packet transmitted; however, you can easily obtain nationwide coverage from almost any cellular phone company.

exam watch

Narrowband solutions provide a low data rate. This can be overcome using spread spectrum, which spreads a signal across multiple frequencies and therefore increases your bandwidth over short distances. Cisco's Aironet products use spread spectrum. Broadband solutions, such as PCS, provide low data rates but can provide a large coverage area. Infrared solutions provide high data rates over very small distances, while satellite connections provide international coverage but have high latency and cost.

Wireless is becoming very popular in today's LANs, since very little cabling is required. Three basic standards are currently in use: 802.11a, 802.11b, and 802.11g, shown in Table 2-2.

Of the three, 802.11b has been deployed the most, with 802.11g just introduced as a standard. One advantage that 802.11b and 802.11g devices have over 802.11a

TABLE 2-2 Wireless Standards

	802.11a	802.11b	802.11g
Data Rate	54 Mbps	11 Mbps	54 Mbps
Frequency	2.4 GHz	5 GHz	2.4 GHz
Compatibility	None	With 802.11g	With 802.11b
Range	25–75 feet	100–150 feet	100–150 feet

> **exam watch** ***Be familiar with the contents of Table 2-2, especially the data rates. The 802.11b standard is commonly called*** *Wi-Fi,* ***even though the term applies to all 802.11 standards.***

is that 802.11b and 802.11g can interoperate, which makes migrating from an all-802.11b network to an 802.11g network an easy and painless process. Note that 802.11g devices are compatible with 802.11b devices (but not vice versa) and that 802.11a devices are *not* compatible with the other two standards. Also note that the speeds listed in Table 2-2 are optimal speeds based on the specifications—the actual speeds that you might achieve in a real network vary according to the number of devices you have, the distance that they are from the base station, and any physical obstructions or interference that might exist.

One of the biggest problems of wireless networks is security. Many wireless networks use Wired Equivalency Privacy (WEP) for security. This is an encryption protocol that uses 40-bit keys, which is weak by today's standards. Many vendors use 128-bit keys to compensate this weakness; however, weaknesses have been found in this protocol, and WEP is used with other security measures to provide a more secure wireless network. The 802.1x/EAP (Extensible Authentication Protocol) is used to provide authentication services for devices: it authenticates devices to an authentication server (typically a RADIUS server) before the device is allowed to participate in the wireless network. Cisco has developed an extension to this called LEAP, or lightweight EAP. LEAP centralizes both authentication and key distribution (for encryption) to provide scalability for large wireless deployments.

Devices

Table 2-3 is a reminder of the devices that function at various OSI Reference Model layers.

TABLE 2-3

Devices and the Layers at Which They Function

Layer	Name of Layer	Device
3	Network	Routers
2	Data link	Switches, bridges, NICs
1	Physical	Hubs

CERTIFICATION OBJECTIVE 2.02

Data Link Layer

Layer 2 of the OSI Reference Model is the data link layer. This layer is responsible for defining the format of layer-2 frames as well as the mechanics of how devices communicate with each other over the physical layer. Here are the components the data link layer is responsible for:

- Defining the Media Access Control (MAC) or hardware addresses
- Defining the physical or hardware topology for connections
- Defining how the network layer protocol is encapsulated in the data link layer frame
- Providing both connectionless and connection-oriented services

Normally, the data link layer does not provide connection-oriented services (ones that do error detection *and* correction). However, in environments that use SNA (Systems Network Architecture) as a data link layer protocol, SNA can provide sequencing and flow control to ensure the deliver of data link layer frames. SNA was developed by IBM to help devices communicate in LAN networks (predominantly Token Ring) at the data link layer. In most instances, it will be the transport layer that provides for reliable connections.

Make sure to remember that the primary function of the data link layer is to regulate how two networking devices connected to the same media type communicate with each other. If the devices are on different media types, the network layer typically plays a role in the communication of these devices.

Data Link Layer Addressing

The data link layer uses MAC, or hardware, addresses for communication. For LAN communications, each machine on the same connected media type needs a unique MAC address. A MAC address is 48 bits in length and is represented as a hexadecimal number. Represented in hex, it is 12 characters in length. To make it easier to read, the MAC address is represented in a dotted hexadecimal format, like this: FFFF.FFFF.FFFF. Since the MAC addresses uses hexadecimal numbers, the values used range from 0–9 and A–F, giving you a total of 16 values for a single digit. For example, a hexadecimal value of A would be 10 in decimal. There are other types of data link layer addressing besides MAC addresses. For instance, Frame Relay uses Data Link Connection Identifiers (DLCIs). I'll discuss DLCIs in more depth in Chapter 16.

The first six digits of a MAC address are associated with the vendor, or maker, of the NIC. Each vendor has one or more unique sets of six digits. These first six digits are commonly called the *organizationally unique identifier* (OUI). For example, one of Cisco's OUI values is *0000.0C*. The last six digits are used to uniquely represent the NIC within the OUI value. Theoretically, each NIC has a unique MAC address. In reality, however, this is probably not true. What is important for your purposes is that each of your devices has a unique MAC address on its NIC within the same *physical* or *logical* segment. A logical segment is a virtual LAN (VLAN) and is referred to as a broadcast domain, which is discussed in Chapter 8. Some devices allow you to change this hardware address, while others won't.

exam watch

The first six hexadecimal digits of a MAC address form the OUI. MAC addresses only need to be unique in a broadcast domain, which includes all layer-2 connected collision domains. You can have the same MAC address in different broadcast domains (virtual LANs).

Each data link layer frame contains two MAC addresses: a source MAC address of the machine creating the frame and a destination MAC address for the device or devices intended to receive the frame. There are three general types of addresses at the data link layer, shown in Table 2-4. A source MAC address is an example of a unicast address—only one device can create the frame. However, destination MAC addresses can be any of the addresses listed in Table 2-4. The destination MAC address in the data link layer frame helps the other NICs connected to the segment to figure out if they need to process the frame when they receive it or to ignore it. The following sections covers each of these address types in more depth.

TABLE 2-4 Data Link Address Types

Address Type	Description
Unicast	Represents a single device on a segment
Broadcast	Represents every device on a segment
Multicast	Represents a group of devices on a segment

Unicast

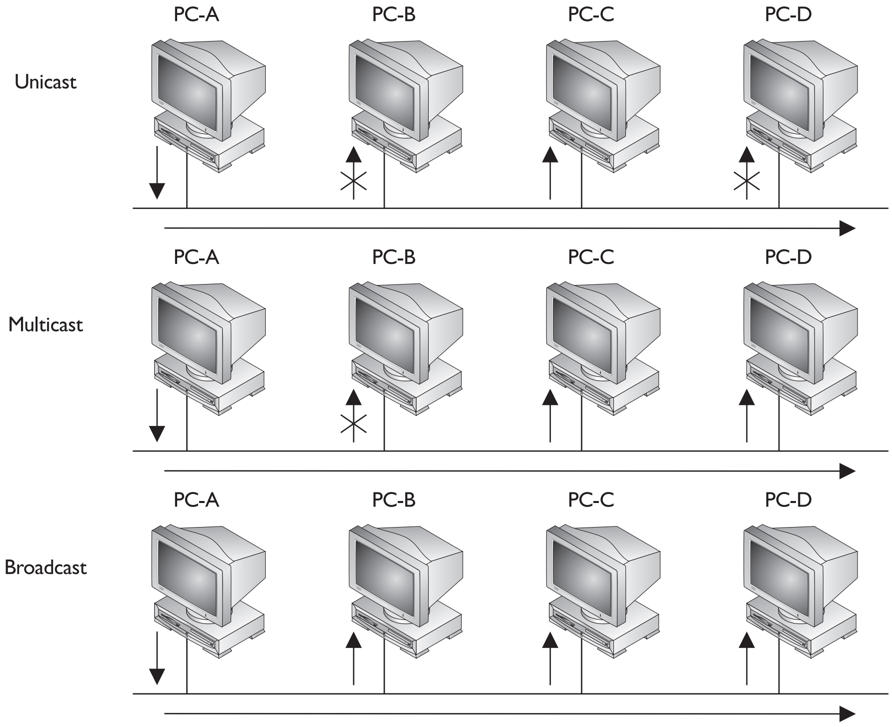
A frame with a destination *unicast* MAC address is intended for just one device on a segment. The top part of Figure 2-2 shows an example of a unicast. In this example, PC-A creates an Ethernet frame with a destination MAC address that contains PC-C's address. When PC-A places this data link layer frame on the wire, all the devices on the segment receive. Each of the NICs of PC-B, PC-C, and PC-D examine the destination MAC address in the frame. In this instance, only PC-C's NIC will process the frame, since the destination MAC address in the frame matches the MAC address of its NIC. PC-B and PC-D will ignore the frame.

Multicast

Unlike a unicast address, a *multicast* address represents a group of devices on a segment. The multicast group can contain anywhere from no devices to every device on a segment. One of the interesting things about multicasting is that the membership of a group is dynamic—devices can join and leave as they please. The detailed process of multicasting is beyond the scope of this book, however.

FIGURE 2-2

MAC address types



The middle portion of Figure 2-2 shows an example of a multicast. In this example, PC-A sends a data link layer frame to a multicast group on its segment. Currently, only PC-A, PC-C, and PC-D are members of this group. When each of the PCs receives the frame, its NIC examines the destination MAC address in the data link layer frame. In this example, PC-B ignores the frame, since it is not a member of the group. However, PC-C and PC-D will process the frame.

Broadcast

A *broadcast* is a data link layer frame that is intended for every networking device on the same segment. The bottom portion of Figure 2-2 shows an example of a broadcast. In this example, PC-A puts a broadcast address in the destination field of the data link layer frame. For MAC broadcasts, all of the bit positions in the address are enabled, making the address FFFF.FFFF.FFFF in hexadecimal. This frame is then placed on the wire. Notice that in this example, when PC-B, PC-C, and PC-D receive the frame, they *all* process it.

Broadcasts are mainly used in two situations. First, broadcasts are more effective than unicasts if you need to send the same information to every machine. With a unicast, you would have to create a separate frame for each machine on the segment; with a broadcast, you could accomplish the same thing with one frame. Second, broadcasts are used to discover the unicast address of a device. For instance, when you turn on your PC, initially, it doesn't know about any MAC addresses of any other machines on the network. A broadcast can be used to discover the MAC addresses of these machines, since they will all process the broadcast frame. In IP, the Address Resolution Protocol (ARP) uses this process to discover another device's MAC address. ARP is discussed in Chapter 3.

Ethernet

Ethernet is a LAN media type that functions at the data link layer. Ethernet uses the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) mechanism to send information in a shared environment. Ethernet was initially developed with the idea that many devices would be connected to the same physical piece of wiring. The acronym CSMA/CD describes the actual process of how Ethernet functions.

In a traditional, or hub-based, Ethernet environment, only one NIC can successfully send a frame at a time. All NICs, however, can simultaneously listen to information on the wire. Before an Ethernet NIC puts a frame on the wire, it will first *sense* the wire to ensure that no other frame is currently on the wire. If the cable uses copper, the NIC can detect this by examining the voltage levels on the wire. If the cable is fiber, the NIC can also detect this by examining the light frequencies on the wire. The NIC must go through this sensing process, since the Ethernet medium supports *multiple access*—another NIC might already have a frame on the wire. If the NIC doesn't sense a frame on the wire, it will go ahead and transmit its own frame; otherwise, if there is a frame on the wire, the NIC will wait for the completion of the transmission of the frame on the wire and then transmit its own frame.

If two or more machines simultaneously sense the wire and see no frame, and each places its frame on the wire, a *collision* will occur. In this situation, the voltage levels on a copper wire or the light frequencies on a piece of fiber get messed up. For example, if two NICs attempt to put the same voltage on an electrical piece of wire, the voltage level will be different than if only one device does so. Basically, the two original frames become unintelligible (or undecipherable). The NICs, when they place a frame on the wire, examine the status of the wire to ensure that a collision does not occur: this is the *collision detection* mechanism of CSMA/CD.

If the NICs see a collision for their transmitted frames, they have to resend the frames. In this instance, each NIC that was transmitting a frame when a collision occurred creates a special signal, called a jam signal, on the wire, waits a small random time period, and senses the wire again. If no frame is currently on the wire, the NIC will then retransmit its original frame. The time period that the NIC waits is measured in microseconds, a delay that can't be detected by a human. Likewise, the time period the NICs wait is random to help ensure a collision won't occur again when these NICs retransmit their frames.

The more devices you place on a segment, the more likely you are to experience collisions. If you put too many devices on the segment, too many collisions will occur, seriously affecting your throughput. Therefore, you need to monitor the number of collisions on each of your network segments. The more collisions you experience, the less throughput you'll get. Normally, if your collisions are less than one percent of your total traffic, you are okay. This is not to say that collisions are *bad*—they are just one part of how Ethernet functions.

Because Ethernet experiences collisions, networking devices that share the same medium (are connected to the same physical segment) are said to belong to the same *collision*, or *bandwidth*, *domain*. This means that, for better or worse, traffic generated by one device in the domain can affect other devices. Chapter 7 discusses how bridges and switches can be used to solve collision and bandwidth problems on a network segment.

exam watch

Make sure you understand the mechanics of Ethernet: CSMA/CD. No device has priority over another device. If two devices transmit simultaneously, a collision occurs. When this happens, a jam signal is generated and the devices try to retransmit after waiting a random period.

IEEE's Version of Ethernet

There are actually two variants of Ethernet: IEEE's implementation and the DIX implementation. Ethernet was developed by three different companies in the early 1980s: Digital, Intel, and Xerox, or DIX for short. This implementation of Ethernet has evolved over time; its current version is called Ethernet II. Devices running TCP/IP typically use the Ethernet II implementation.

The second version of Ethernet was developed by IEEE and is standardized in the IEEE 802.2 and 802.3 standards. IEEE has split the data link layer into two components: MAC and LLC. These components are described in Table 2-5. The top part of the data link layer is the LLC, and its function is performed in software. The bottom part of the data link layer is the MAC, and its function is performed in hardware.

The LLC performs its multiplexing by using Service Access Point (SAP) identifiers. When a network layer protocol is encapsulated in the 802.2 frame, the protocol of the network data is placed in the SAP field. When the destination receives the frame, it examines the SAP field to determine which upper-layer network layer protocol should process the frame. This allows the destination network device to differentiate

TABLE 2-5 IEEE Ethernet Components

Data Link Layer	Name	IEEE Standard	Description
Top part	Logical Link Control (LLC)	802.2	Defines how to multiplex multiple network layer protocols in the data link layer frame. LLC is performed in *software*.
Bottom part	MAC	802.3	Defines how information is transmitted in an Ethernet environment, and defines the framing, MAC addressing, and mechanics as to how Ethernet works. MAC is performed in *hardware*.

between TCP/IP and IPX network layer protocols that are being transmitted across the data link layer connection. Optionally, LLC can provide sequencing and flow control to provide a reliable service, as TCP does at the transport layer. However, most data link layer implementations of Ethernet don't use this function—if a reliable connection is needed, it is provided by either the transport or application layer.

IEEE 802.3 As mentioned earlier, IEEE 802.3 is responsible for defining the framing used to transmit information between two NICs. A frame standardizes the fields in the frame and their lengths so that every device understands how to read the contents of the frame. The top part of Figure 2-3 shows the fields of an 802.3 frame.

Table 2-6 shows the fields found in the 802.3 frame. The field checksum sequence (FCS) value is used to ensure that when the destination receives the frame, it can verify that the frame was received intact. When generating the FCS value, which is basically a checksum, the NIC takes all of the fields in the 802.3 frame, except the FCS field, and runs them through an algorithm that generates a four-byte result, which is placed in the FCS field.

When the destination receives the frame, it takes the same fields and runs them through the same algorithm. The destination then compares its four-byte output with what was included in the frame by the source NIC. If the two values don't match, then the frame is considered bad and is dropped. If the two values match, then the frame is considered good and is processed further.

IEEE 802.2 IEEE 802.2 (LLC) handles the top part of the data link layer. There are two types of IEEE 802.2 frames: Service Access Point (SAP) and Subnetwork Access Protocol (SNAP). These 802.2 frames are encapsulated (enclosed) in an 802.3 frame when being sent to a destination. Where 802.3 is used as a transport to get the 802.2 frames to other devices, 802.2 is used to define which network layer

FIGURE 2-3

Ethernet 802.3 and Ethernet II

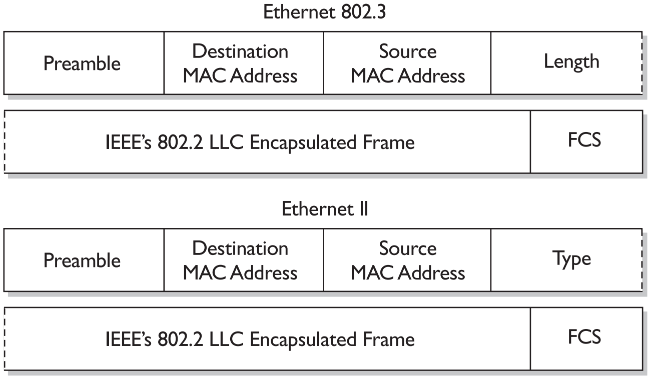


TABLE 2-6 Fields in the 802.3 Frame

Field	Length in Bytes	Description
Preamble	8	Identifies the beginning of the 802.3 frame
Destination MAC address	6	Is the MAC address that the frame is to be sent to
Source MAC address	6	Is the MAC address of the source of the frame
Length	2	Defines the length of the frame from this point to the checksum at the end of the frame
Data	Variable	Is the 802.2 LLC encapsulated frame
FCS (Field Checksum Sequence)	4	Is a checksum (CRC, cyclic redundancy check) that is used to ensure that the frame is received by the destination error-free

protocol created the data that the 802.2 frame will include. In this sense, it serves as a multiplexing function: it differentiates between TCP/IP, IPX, AppleTalk, and other network-layer data types. Figure 2-4 shows the two types of 802.2 frames.

Table 2-7 lists the fields found in an 802.2 SAP frame.

When a destination NIC receives an 802.3 frame, the NIC first checks the FCS to verify that the frame is valid and then checks the destination MAC address in the 802.3 frame to make sure that it should process the frame (or ignore it). The MAC sublayer strips off the 802.3 frame portion and passes the 802.2 frame to the LLC sublayer. The LLC examines the destination SAP value to determine which upper-layer protocol should have the encapsulated data passed to it. Here are some examples of SAP values: IP uses 0x06 (hexadecimal) and IPX uses 0x0E. If the LLC sees 0x06 in the SAP field, it passes the encapsulated data up to the TCP/IP protocol stack running on the device.

FIGURE 2-4

SAP and SNAP

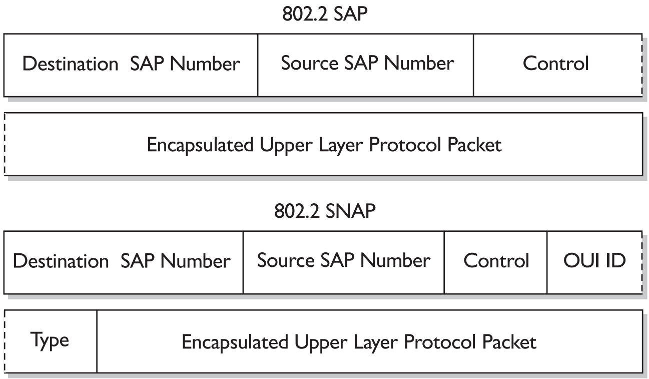


TABLE 2-7 802.2 SAP Fields

Field	Length in Bytes	Description
Destination SAP number	1	Identifies the network layer protocol that this is to be sent to
Source SAP number	1	Identifies the network layer protocol that originated this data
Control field	1–2	Determines the fields that follow this field
Data	Variable	This contains the upper-layer network layer packet

The second frame type supported by 802.2 is SNAP, which is shown in the bottom portion of Figure 2-4. As you can see from this frame, there is one additional field: *type*. Table 2-8 explains the 802.2 SNAP fields.

One of the issues of the original SAP field in the 802.2 SAP frame is that even though it is eight bits (one byte) in length, only the first six bits are used for identifying upper-layer protocols, which allows up to 64 protocols. Back in the 1980s, there were many more protocols than 64, plus there was an expectation that more protocols would be created. SNAP overcomes this limitation without having to change the length of the SAP field.

To indicate a SNAP frame, the SAP fields are set to hexadecimal *0xAA*, the control field is set to *0x03*, and the OUI field is set to *0x0*. The *type* field identifies the upper-layer protocol that is encapsulated in the payload of the 802.2 frame. Since a SAP frame can identify only 64 protocols, the type field was made two bytes in

TABLE 2-8 802.2 SNAP Fields

Field	Length in Bytes	Description
Destination SAP number	1	This is set to **0xAA** to signify a SNAP frame
Source SAP number	1	This is set to **0xAA** to signify a SNAP frame
Control field	1-2	This is set to **0x03** to signify a SNAP frame
OUI ID	3	This value varies by vendor but is set to **0x0** to signify a SNAP frame
Type	2	This indicates the upper-layer protocol that is contained in the data field
Data	Variable	This contains the upper-layer network layer packet

exam watch

802.2 uses a SAP or SNAP field to differentiate between encapsulated layer-3 payloads. With a SNAP frame, the SAP fields are set to 0xAA and the type field is used to indicate the layer-3 protocol.

length, which theoretically allows the support of up to 65,536 protocols! AppleTalk is an example of a protocol that uses an 802.2 SNAP frame.

Note that concerning 802.2 there are other data link layer protocols for the LAN besides Ethernet, including Token Ring and FDDI. IEEE's 802.2 standard supports these sublayer standards at the MAC layer. Token Ring is specified in IEEE's 802.5 standard, and FDDI is specified in an ANSI standard. This book only focuses on Ethernet.

Ethernet II's Version of Ethernet

Ethernet II is the original Ethernet frame type. Ethernet II and 802.3 are very similar: they both use CSMA/CD to determine their operations. Their main difference is the frames used to transmit information between NICs. The bottom part of earlier Figure 2-3 shows the fields in an Ethernet II frame. Here are the two main differences between an Ethernet II and IEEE:

- Ethernet II does not have any sublayers, while IEEE 802.2/3 have two: LLC and MAC.
- Ethernet II has a *type* field instead of a length field (used in 802.3). IEEE 802.2 defines the type for IEEE Ethernet.

exam watch

Be able to compare and contrast the two versions of Ethernet: Ethernet II and 802.2/3.

If you examine the IEEE 802.3 frame and the Ethernet II frame, you can see that they are very similar. NICs differentiate them by examining the value in the type field for an Ethernet II frame and the value in the length field in the IEEE 802.3 frame. If the value is greater than 1500, then the frame is an Ethernet II frame. If the value is 1500 or less, the frame is an 802.3 frame.

Both versions of Ethernet can coexist in the same network. However, because of the frame differences between the two types, a NIC running only 802.3 will discard any Ethernet II frames and vice versa.

Ethernet Physical Layer Properties

Many physical layer standards define the physical properties of an Ethernet implantation. One of the most common is IEEE's 802.3 10Mb. Table 2-9 shows some of the 10Mb standards.

Ethernet supports a bus topology—physical or logical. In a bus topology, every device is connected to the same piece of wire and all devices see every frame. For example, 10Base5 uses one long, thick piece of coaxial cable. NICs tap into this wire using a device called a vampire tap. With 10Base2, the devices are connected together by many pieces of wire using T-taps: one end of the T-tap connects to the NIC and the other two connect to the two Ethernet cables that are part of the bus. With 10BaseT, all devices are connected to a hub, where the hub provides a logical bus topology. All of these 10Mb Ethernet solutions support only half-duplex: they can send or receive. They cannot do both simultaneously. Duplexing is discussed in more depth in Chapter 7.

exam watch

Half-duplex connections allow devices to either send or receive and experience collisions. Full-duplex connections require a point-to-point connection between two devices. With this type of connection, both devices can simultaneously send and receive without any collisions occurring.

Ethernet 10Base2 and 10Base5 haven't been used in years because of the difficulty in troubleshooting network problems. And many 10BaseT networks have been supplanted by higher-speed Ethernet solutions, like Fast Ethernet and Gigabit

TABLE 2-9 10Mb Ethernet Properties

Ethernet Type	Distance Limitation	Cable Type	Interface Type	Physical Topology	Logical Topology
10Base5	500 meters	Thick coaxial cable—50 ohm *(thicknet)*	AUI	Bus	Bus
10Base2	185 meters	Thin coaxial cable *(thinnet)*	BNC	Bus	Bus
10BaseT	100 meters	Unshielded twisted pair (UTP) cabling (CAT-3, -4, -5)	RJ-45	Star (Hub)	Bus

TABLE 2-10 100Mb Ethernet Properties

Ethernet Type	Distance Limitation	Cable Type	Cabling	Physical Topology	Logical Topology
100BaseTX	100 meters	UTP CAT-5	RJ-45	Star (Hub)	Bus
100BaseFX	400 meters half-duplex, 2000 meters full-duplex	MMF 62.5/125 micron with SC and ST connectors	RJ-45	Star (Hub)	Bus
100BaseT4	100 meters	UTP CAT-3,4,5	RJ-45	Star (Hub)	Bus

Ethernet. Fast Ethernet and Ethernet use the same frame types and support the same CSMA/CD operation. However, there are two main differences between the two: Fast Ethernet supports 100 Mbps speeds and the physical layer is implemented differently. Table 2-10 shows the different implementations of Fast Ethernet. Fast Ethernet supports both half- and full-duplex connections. With full-duplex connections, a device can send *and* receive simultaneously but requires a point-to-point connection that doesn't involve a hub.

Gigabit Ethernet is defined in IEEE 802.3z. To achieve 1Gbps speeds, IEEE adopted ANSI's X3T11 Fiber Channel standard for the physical layer implantation. The physical layer is different from Ethernet and Fast Ethernet in that it uses an 8B/10B encoding scheme to code the physical layer information when transmitting it across the wire. Table 2-11 shows the different implementations of 1Gbps. There is also a 10Gbps implementation of Ethernet that only runs across fiber. This standard is currently in the development process.

Table 2-12 compares the different cable types.

TABLE 2-11 1 Gbps Ethernet Properties

Ethernet Type	Distance Limitation	Cable Type
1000BaseCX	25 meters	Shielded twisted pair (STP) copper
1000BaseLX	3–10 kilometers	SMF
1000BaseSX	220 meters	MMF
1000BaseT	100 meters	CAT-5E and CAT-6 UTP
1000BaseZX	100 meters	SMF

TABLE 2-12 Cable Type Comparisons

Cable	Distance	Data Rates	Comparison
UTP	100 meters	10–1000Mbps	Is easy to install but is susceptible to interference
STP (Shielded Twisted Pair)	100 meters	10–100Mbps	Is difficult to install
Coaxial	500 meters	10–100Mbps	Is easy to install but is difficult to troubleshoot
Fiber	10 kilometers	10Mbps–100Gbps	Is difficult and expensive to install, difficult to troubleshoot, but can span very long distances and is not susceptible to interference

Data Link Devices: Bridges

Bridges are data link layer devices that switch frames between different layer-2 segments. They perform their switching in software, and their switching decisions are based on the destination MAC address in the header of the data link layer frame.

Bridges perform three main functions:

- They learn where devices are located by placing the MAC address of a device and the identifier of the port it is connected to in a port address table.
- They forward traffic intelligently, drawing on information they have in their port address table.
- They remove layer-2 loops by running the Spanning Tree Protocol (STP).

Actually, these three functions are implemented in bridges that perform transparent bridging. There are other types of bridging, including translational bridging, source route bridging, source route transparent bridging, and source route translational bridging. However, this book only focuses on transparent bridging. The following sections introduce you to bridging; Chapter 7 goes into more depth about this subject.

exam watch

The three main functions of a bridge are learn, forward, and remove loops.

Learning Function

One of the three functions of a bridge is to learn which devices are connected to which ports of the bridge. The bridge then uses this information to switch frames intelligently. When a bridge receives a frame, it reads the source MAC address in the frame and compares it to a local MAC address table, called a port address table. If the address is not already in this table, the bridge adds the address and the port identifier on which the frame was received. If the address is already in the table, the bridge resets the timer for the table entry. Entries in the table remain there as long as the bridge sees traffic from them; otherwise, the bridge ages out the old entries to allow room for newer ones.

Forwarding Function

The second function of a bridge is to intelligently forward traffic. In order to do this, the bridge uses the port address table to help it find where destinations are located. When a frame is received on a port, the bridge first performs its learning function and then performs its forwarding function. The bridge examines the destination MAC address in the frame header and looks for a corresponding entry in the port address table. If the bridge finds a matching entry, the bridge forwards the frame out of the specified port. If the port is the same port on which the frame was received (the source and destination are connected to the same port), the bridge drops the frame. If the bridge doesn't find an entry, or if the destination MAC address is a broadcast or multicast address, the bridge *floods* the frame out all of the remaining ports.

exam watch

Remember that these three types of traffic are always flooded: unknown unicast addresses, broadcasts, and multicasts.

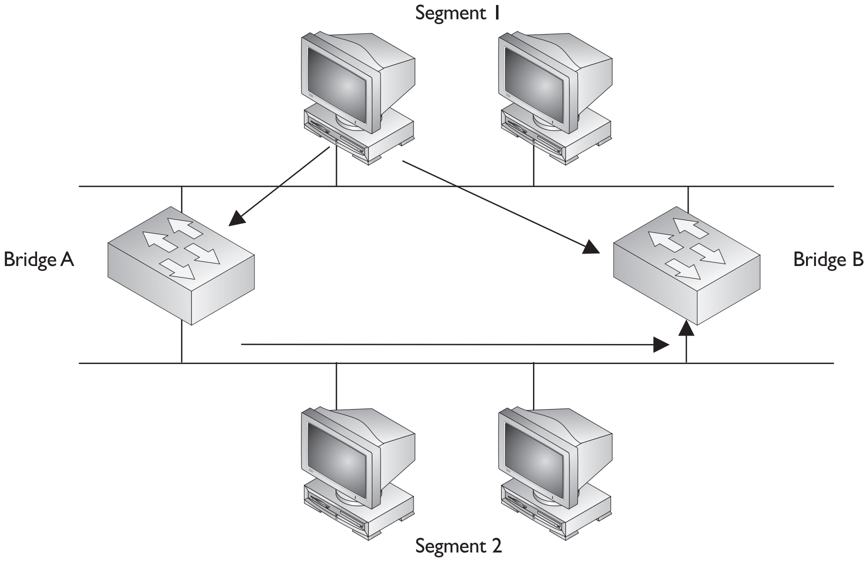
Removing Loops

The third function of a bridge is to remove layer-2 loops. To see the problem that layer-2 loops can cause, consider Figure 2-5. One advantage of using two bridges to connect two segments together, as is shown in Figure 2-5, is that you have redundancy.

But these loops also create problems. For instance, a bridge always floods traffic that has a destination address that is an unknown unicast, broadcast, or multicast address. And this traffic will continually circle around the loop—possibly forever. For example, in Figure 2-5, assume that a PC generates a broadcast on Segment1. When BridgeA and BridgeB receive the broadcast, they flood it out all of their remaining ports. This means that the same broadcast will appear twice on Segment2. Each bridge sees the other's broadcast on Segment2 and forwards this to Segment1. And this process will go on ad infinitum. This process not only wastes bandwidth on your LAN segments but also affects the CPU cycles of all devices on these segments, since all NICs will accept the broadcast and pass it up the protocol stack for further processing.

FIGURE 2-5

Layer-2 loops and redundancy



The Spanning Tree Protocol (STP) is used to remove loops in your layer-2 network. When STP runs, one of the ports of the bridges in a loop is disabled in software. In Figure 2-5, this is the port on BridgeB that is connected to Segment2. Any user traffic is ignored if it is received on this port and is not forwarded out of this port. Going back to our broadcast example, if a PC on Segment1 generated a broadcast, both bridges, again, would receive it. BridgeA would flood the broadcast to Segment2, but BridgeB would not, since the port is in a *blocked* state. STP is discussed in much more depth in Chapter 7.

Problems That Bridges Solve

Bridges are used to solve collision and bandwidth problems. Each port connected to a bridge is a separate collision domain. When a frame is pulled into a port on a bridge, the bridge checks the frame's FCS, and if the FCS if valid, the frame is forwarded out of a destination port or ports. Basically, the bridge is creating the illusion that all the physical segments that it is connected to are actually one large logical segment. All devices connected to this "logical" segment are in the same broadcast domain—this makes sense because bridges flood

exam watch

STP is used to remove layer-2 loops.

broadcasts. Note that if you are having problems with large amounts of broadcasts, bridges will not solve these problems.

Data Link Devices: Switches

Switches, like bridges, operate at the data link layer. The three main functions of a bridge are also true of a switch: they learn, forward, and remove loops. However, switches have many more features than bridges; for instance, they make their switching decisions in hardware by using application-specific integrated circuits (ASICs). ASICs are specialized processors built to perform very few specific tasks. Because they do only a few things, ASICs are much more cost-effective than a generic processor, like the one found in your PC. Cisco, like most networking vendors, extensively uses ASICs throughout its switching products. Chapter 7 continues the discussion of the differences between bridges and switches.

exam watch

Bridges, as well as switches, are used to solve bandwidth and collision problems. Routers, at the network layer, can also perform this function, but they cost more than bridges or switches.

CERTIFICATION OBJECTIVE 2.03

Network Layer

Layer 3 of the OSI Reference Model is the network layer. This layer is responsible for three main functions:

- Defines logical addresses used at layer-3
- Finds paths, based on the network numbers of logical addresses, to reach destination devices
- Connects different data link types together, such as Ethernet, FDDI, Serial, and Token Ring

The following sections cover the network layer in more depth.

Layer-3 Addressing

Many protocols function at the network layer: AppleTalk, DECnet, IP, IPX, Vines, XNS, and others. Each of these protocols has its own method of defining logical addressing. Correct assignment of these addresses on devices across your network allows you to build a hierarchical design that can scale to very large sizes. This provides an advantage over layer-2 addresses, which use a flat design and are not scalable.

All layer-3 addressing schemes have two components: network and host (or node). Each segment (physical or logical) in your network needs a unique network number. Each host on these segments needs a unique host number from within the assigned network number. The combination of the network and host number assigned to a device provides a unique layer-3 address throughout the entire network. For example, if you had 500 devices in your network that were running IP, each of these devices would need a unique IP layer-3 address.

This process is different with MAC addresses, which are used at layer-2. MAC addresses need to be unique only on a physical (or logical) segment. In other words, within the same broadcast domain, all of the MAC addresses must be unique. However, MAC addresses do *not* need to be unique between two *different* broadcast domains. An example of this appears later in this chapter.

To understand the components of layer-3 addresses, let's look at a few examples. TCP/IP addresses are 32 bits in length. To make these addresses more readable, they are broken up into four bytes, or *octets*, where any two bytes are separated by a period. This is commonly referred to as dotted decimal notation. Here's a simple example of an IP address: 10.1.1.1. An additional value, called a *subnet mask,* determines the boundary between the network and host components of an address. When comparing IP addresses to other protocols' addressing schemes, IP is the most complicated. IP addressing is thoroughly covered in Chapter 3.

Most other protocols have a much simpler format. For example, IPX addresses are 80 bits in length. The first 32 bits are always the network number, and the last 48 bits are always the host address. IPX addresses are represented in hexadecimal. Here's an example: ABBA.0000.0000.0001. In this example, ABBA is the network number and 0000.0000.0001 is the host number. Every protocol has its own addressing scheme. However, each scheme always begins with a network component followed by a host component.

Routing Tables

Routers are devices that function at the network layer; they use network numbers to make routing decisions: how to get a packet to its destination. Routers build a *routing* *table*, which contains path information. This information includes the network number, which interface the router should use to reach the network number, the metric of the path (what it costs to reach the destination), and how the router learned about this network number. Metrics are used to weight the different paths to a destination. If there is more than one way to reach the destination, the metric is used as a tie-breaker. The router will put the best metric paths in its routing table.

There are many different types of metrics, such as bandwidth, delay, and hop count. Each routing protocol uses its own metric structure. For instance, IP RIP uses hop count, while Cisco's EIGRP uses bandwidth, delay, reliability, load, and frame size (MTU). Routing and routing metrics are discussed in Chapters 9, 10, and 11.

When a router receives an inbound packet, it examines the destination layer-3 address in the packet header. The router then determines what the network number is in the address and then compares this network number to its routing table entries. If the router finds a match, it forwards the packet out of the destination interface. However, if the router does not find a match, the router *drops* the packet. This is unlike bridges and switches at layer 2: with these devices, unknown unicast destinations are flooded, not dropped.

exam watch

Routers make routing decisions based on the network numbers in layer-3 addresses, like IP addresses. Locations of networks are stored in a routing table.

Advantages of Routers

Because routers operate at a higher layer than the network layer and use logical addressing, they provide many advantages over bridges and switches, including:

- Logical addressing at layer-3 allows you to build hierarchical networks that scale to very large sizes. This is discussed in Chapter 12.
- They contain broadcasts and multicasts. When a broadcast or multicast is received on an interface, it is *not* forwarded to another interface, by default. Routers are used to solve broadcast problems. (Actually, routers also create separate bandwidth and collision domains, but bridges and switches provide a cheaper solution.)
- Routers can typically find a better path to a destination than bridges, since routing protocols support a rich metric structure.
- Routers allow you to connected different media types together, like Ethernet and Token Ring or FDDI and PPP, without any conversion issues.

- Routers can switch packets on the same interface using VLANs. (VLANs are discussed in Chapter 8.)
- Routers have advanced features that allow you to implement Quality of Service using queuing or traffic shaping, filtering traffic using access lists, or protecting traffic using encryption. (Access lists are discussed in Chapter 13.)

By using logical addresses, routers can create a hierarchical network that supports thousands of devices. Bridges and switches, on the other hand, do not support hierarchical addressing: MAC addresses support a flat addressing space. In other words, you can't typically change MAC addresses to fit a specific network layout. Also, since routers use logical addresses, it is much easier to implement policy decisions, such as traffic filtering or quality service, since the decisions are made on logical, more easily handled addresses than the physical addresses that bridges and switches use. For example, since logical addresses support a network component, you could filter an entire network number. To accomplish this with a bridge, you would have to filter each individual device's MAC address within the network segment.

Another problem with layer-2 devices is that they don't operate very well when connecting different media types, Ethernet and Token Ring, for instance. At layer 2, this process is called *translational bridging*. There are many reasons why layer-2 devices have issues translating media types, but the main reason is that since both topologies are layer-2, the bridge has to translate the layer-2 information from the different media types. This is very process-intensive and can create many problems.

For example, Ethernet supports frame sizes up to 1,500 bytes, while Token Ring supports frame sizes up to 16KB in size for 16Mbps speeds. Therefore, if a large Token Ring frame had to be sent to an Ethernet segment, the bridge would have to fragment the information into multiple Ethernet frames. There might also be a speed difference between the media types: Ethernet supports 10Mb while Token Ring supports 4Mbps, 16Mbps, and 100Mbps, and this difference could cause congestion problems on a bridge or switch.

Also, the translation process between frame types is not always easy. For example, some media types order their bits from low-to-high, while others order them high-to-low, which can create translation issues. Fortunately, routers provide a clean solution to this translation process. Routers don't actually translate between different frame or media types; instead, they strip off the layer-2 frame, make a routing decision on the layer-3 packet, and then encapsulate the layer-3 packet in the correct layer-2 frame type for the interface the packet needs to exit. This process is described more thoroughly later in this chapter, in the section "Transferring Information Between Computers."

Another advantage routers have over layer-2 devices is that they contain broadcast problems. When a router receives a broadcast, it processes that broadcast, but by default, it will not forward the broadcast out any of its other ports. This is different from bridges and switches, which flood broadcast traffic. If broadcasts are affecting the bandwidth and performance of your network, you should break up your network into multiple broadcast domains and use a router to route between the different domains. Each broadcast domain in a network needs a unique layer-3 network number.

CERTIFICATION OBJECTIVE 2.04

Transport Layer

The fourth layer of the OSI Reference Model is the transport layer. The transport layer has four main functions:

- It sets up and maintains a session connection between two devices.
- It can provide for the reliable or unreliable delivery of data across this connection.
- It can implement flow control through ready/not ready signals or windowing to ensure one device doesn't overflow another device with too much data on a connection.
- It multiplexes connections, allowing multiple applications to simultaneously send and receive data.

The following sections cover these processes.

Reliable Connections

The transport layer can provide reliable and unreliable transfer of data between networking devices. TCP/IP's Transmission Control Protocol (TCP) is an example of a transport layer protocol that provides a reliable connection. When implementing a reliable connection, sequence numbers and acknowledgments (ACKs) are used. For example, when information is sent to a destination, the destination will acknowledge to the source what information was received. The destination can examine the sequence numbers to determine if anything was missing, as well as put the data back in the correct order, if it arrived out of order, before passing it on to the upper-layer application. If a segment is missing, the destination can request the source to resend the missing information. With some protocol stacks, the destination might have the source resend all of the information, or parts of the information, including the missing parts.

With reliable connections, before a device can send information to another device, a handshake process must take place to establish the connection. Figure 2-6 shows the steps involved. Let's assume that this is TCP setting up a reliable IP connection. In this example, PC-A wants to send data reliably to PC-B. Before this can take place, PC-A must set up a reliable connection to PC-B. The two devices go through a *three-way handshake* to establish the connection. Here are the three steps that occur:

1. The source sends a synchronization (SYN) message to the destination, indicating that the source wants to establish a reliable connection.
2. The destination responds with both an acknowledgment and a synchronization message. The acknowledgment indicates the successful receipt of the source's SYN message, and the destination's SYN message indicates that a connection can be set up. Together, these messages are referred to as SYN/ACK; they are sent together in the same data transfer.
3. Upon receiving the SYN/ACK, the source responds with an ACK message. This indicates to the destination that its SYN was received by the source and that the connection is now complete.

Once the three-way handshake has taken place, data can be transferred across the connection. Because the connection was established first, this type of service is referred to as *connection-oriented*. Remember that this type of connection always goes through a three-way handshake before one device can start sending and receiving information from another.

exam watch

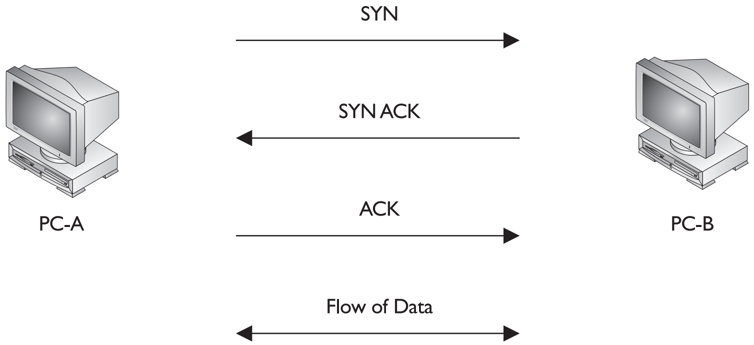
A connection-oriented session goes through a three-way handshake when establishing a session.

Unreliable Connections

One of the issues of connection-oriented services is that they must always go through a three-way handshake before you can transfer data. In some instances, like file transfers, this makes sense, because you want to make sure that all data for the file is transferred successfully. However, in other cases, when you want to send only one piece of information and get a reply back, going through the three-way handshake process adds additional overhead that isn't necessary.

FIGURE 2-6

Setting up a reliable connection: three-way handshake



A DNS query is a good example where using a connection-oriented service doesn't make sense. With a DNS query, a device is trying to resolve a fully qualified domain name to an IP address. The device sends the single query to a DNS server and waits for the server's response. In this process, only two messages are generated: the client's query and the server's response. Because of the minimal amount of information shared between these two devices, it makes no sense to establish a reliable connection first before sending the query. Instead, the device should just send its information and wait for a response. If a response doesn't come back, the application can send the information again or the user can get involved. Again, with DNS, you can configure two DNS servers in the Microsoft Windows operating system. If you don't get a reply from the first server, the application can use the second configured server.

Because no "connection" is built up front, this type of connection is referred to as a *connectionless* service. The TCP/IP protocol stack uses the User Datagram Protocol (UDP) to provide unreliable connections.

Connection Multiplexing

Another function of the transport layer is to set up and maintain connections for the session layer. The information transferred to devices at the transport layer is called a *segment*. Because multiple connections may be established from one device to another device or devices, some type of multiplexing function is needed to differentiate between the various connections. This ensures that the transport layer can send data from a particular application to the correct destination and, when receiving data from a destination, get it to the right application.

To accomplish this feat, the transport layer assigns a unique set of numbers for each connection. These numbers are called *port* or *socket* numbers. There is a source port number and a destination port number for each connection. The destination port numbers assigned by the source device are referred to as well-known port numbers.

The source device uses an appropriate port number in the destination port field to indicate to the destination which application it is trying to access. For example, the TCP/IP protocol stack gives each application a unique port number.

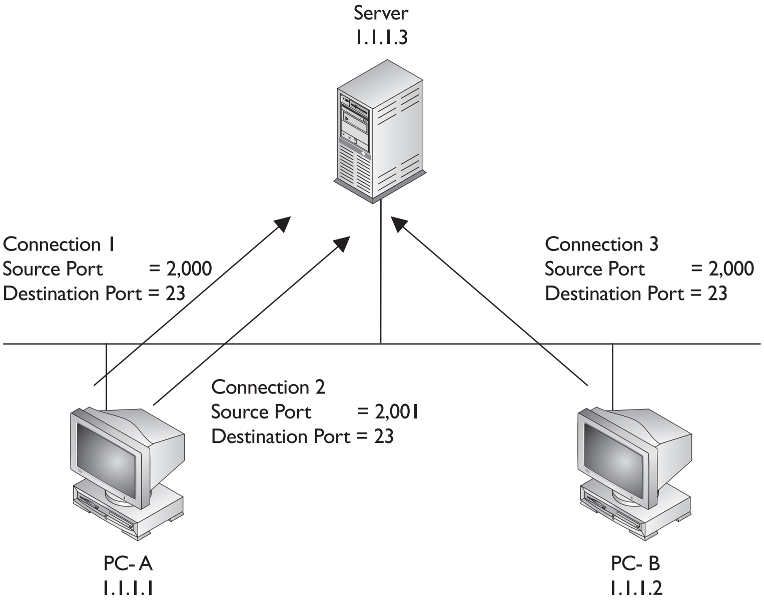
Here are some well-known port numbers used by TCP/IP applications: FTP (20 and 21), telnet (23), SMTP—e-mail (25), DNS (53), TFTP (69), WWW (80), and POP mail (110). With TCP/IP, port numbers from 0–1,023 are well-known port numbers. However, some applications have port numbers higher than these numbers. Actually, TCP/IP uses a 16-bit field for the port number, allowing you to reference up to 65,536 different numbers. Port numbers above 1,023 are used by the source to assign to the connection. Each connection on the source has a unique source port number. This helps the source device differentiate its own connections.

Let's look at an example, shown in Figure 2-7, that uses TCP for multiplexing connections. In this example, PC-A has two telnet connections between itself and the server. You can tell these are telnet connections by examining the destination port number (23). When the destination receives the connection setup request, it knows that the process it should start up is telnet. Also notice that the source port number is *different* for each of these connections (2,000 and 2,001). This allows both the PC and the server to differentiate between the two separate telnet connections. This is a simple example of multiplexing connections.

Of course, if more than one device is involved, things become more complicated. In the example shown in Figure 2-7, PC-B also has a connection to the server. This

FIGURE 2-7

Multiplexing connections



connection has a source port number of 2,000 and a destination port number of 23—another telnet connection. This brings up an interesting dilemma. How does the server differentiate between PC-A's connection that has port numbers 2,000/23 and PC-B's, which has the same? Actually, the server uses not only the port numbers at the transport layer to multiplex connections, but also the *layer-3* addresses of the devices connected to these connections. In this example, notice that PC-A and PC-B have *different* layer-3 addresses: 1.1.1.1 and 1.1.1.2 respectively.

As you can see from this example, no matter where the connections are coming from, or how many connections a device has to deal with, the device can easily differentiate between the connections by examining the source and destination port numbers as well as the layer-3 addresses.

exam watch

The transport layer uses source and destination port numbers and layer-3 addresses to perform multiplexing of connections.

Flow Control

Another function of the transport layer is to provide optional flow control. Flow control is used to ensure that networking devices don't send too much information to the destination, overflowing its receiving buffer space, and causing it to drop the sent information. Overflow is not good because the source will have to resend all of the information that was dropped. The transport layer can use two basic flow control methods:

- Ready/not ready signals
- Windowing

exam watch

The purpose of flow control is to ensure the destination doesn't get overrun by too much information sent by the source.

Reading/Not Ready Signals

With *ready/not ready signals*, when the destination receives more traffic than it can handle, it can send a *not ready* signal to the source, indicating that the source should stop transmitting data. When the destination has a chance to catch up and process the source's information, the destination responds back with a *ready* signal. Upon receiving the ready signal, the source can resume the sending of data.

There are two problems with the use of ready/not ready signals to implement flow control. First, the destination may respond to the source with a not ready signal when its buffer fills up. While this message is on its way to the source, the source is *still sending* information to the destination, which the destination will probably have to drop because its buffer space is full. The second problem with the use of these signals is that once the destination is ready to receive more information, it must first send a ready signal to the source, which must receive it before more information can be sent. This causes a delay in the transfer of information. Because of these two inefficiencies with ready/not ready signals, they are not commonly used to implement flow control.

Windowing

Windowing is a much more sophisticated method of flow control than using ready/not ready signals. With windowing, a window size is defined that specifies how many pieces of information can be sent before the source has to wait for an acknowledgment (ACK) from the destination. Once the ACK is received, the source can send the next batch of information (up to the maximum defined in the window size).

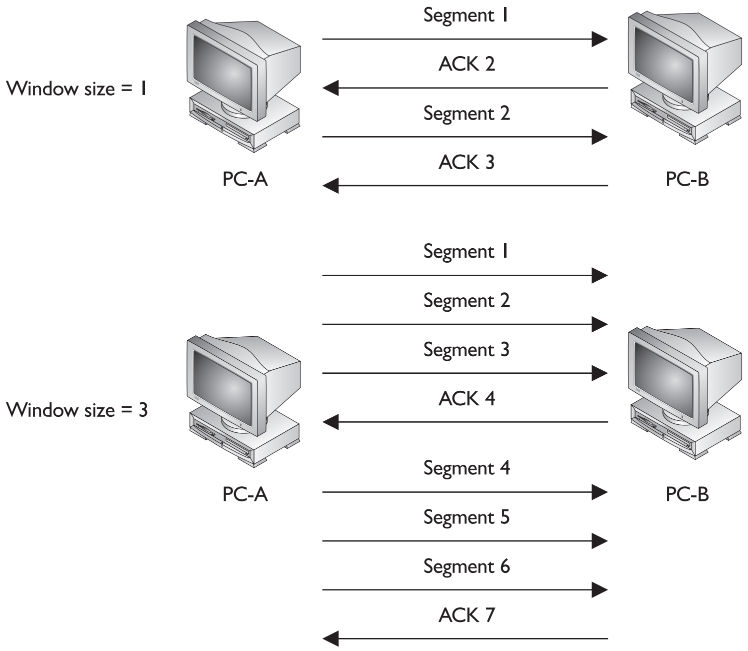
Windowing accomplishes two things. First, flow control is enforced, based on the window size. In many implementations, the window size is dynamically negotiated up front and can be renegotiated during the lifetime of the connection. This ensures that the most optimal window size is used to send information without having the destination drop anything. Second, through the windowing process, the destination tells the source what was received. This indicates to the source if any information was lost along the way to the destination and allows the source to resend any missing information. This provides reliability for a connection as well as better efficiency than ready/not ready signals provide. Because of these advantages, most connection-oriented transport protocols, like TCP/IP's TCP, use windowing to implement flow control.

The window size chosen for a connection impacts its efficiency and throughput in defining how many segments (or bytes) can be sent before the source has to wait for an acknowledgment. Figure 2-8 illustrates the importance of the size used for the window.

The top part of the figure shows the connection using a window size of 1. In this instance, the source sends one segment with a sequence number (in this case 1) and then waits for an acknowledgment from the destination. Depending on the transport protocol, there are different ways the destination can send the acknowledgment: it can send back a list of the sequence numbers of the segments it received, or it can send back the sequence number of the next segment it expects. TCP uses the latter method, which is shown in Figure 2-8. The acknowledgment from the destination has a number 2 in it. This tells the source that it can go ahead and send segment 2. Again, when the destination receives this segment, since the window size is 1, the

FIGURE 2-8

Window sizes affect efficiency.



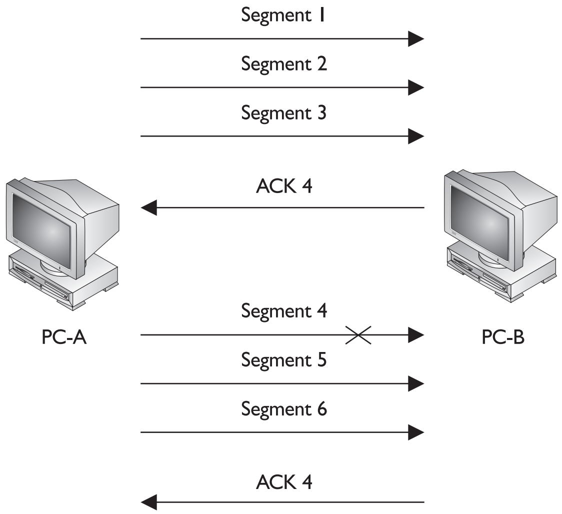
destination will immediately reply with an acknowledgment, indicating the receipt of this segment. In this example, the destination acknowledges back 3, indicating that segment 3 can be sent.

As you can see, with a window size of 1, the flow control process is not very quick or very efficient. Let's look at an example where the window size is 3, as is illustrated at the bottom of Figure 2-8. With a window size of 3, the source can send three segments. Once they are sent (each with its own unique sequence number: 1, 2, and 3), the source must wait for an acknowledgment. In this instance, the destination sends an acknowledgment back with the number 4 in it, indicating that the fourth segment is expected next. The source can then proceed to send segments 4, 5, and 6 and then wait for the destination's acknowledgment. In this instance, having a larger window size is more efficient: only one acknowledgment is required for every three segments that are sent. Therefore, the larger the window size, the more efficient the transfer of information becomes.

However, this is not always the case. For example, let's assume that one segment gets lost on its way to the destination, as is shown in Figure 2-9. In this example, the window size is 3. PC-A sends its first three segments, which are successfully received by PC-B. PC-B acknowledges the next segment it expects, which is 4. When PC-A

FIGURE 2-9

Lost segments and retransmissions



receives this acknowledgment, it sends segments 4, 5, and 6. For some reason, segment 4 becomes lost and never reaches the destination, but segments 5 and 6 do. Remember that the destination is keeping track of what was received: 1, 2, 3, 5, and 6. In this example, the destination sends back an acknowledgment of 4, indicating that segment 4 is expected next.

At this point, how PC-A reacts depends on the transport layer protocol that is used. Here are some possible options:

- PC-A understands that only segment 4 was lost and therefore resends segment 4. It then sends segments 7 and 8, filling up the window size.
- PC-A doesn't understand what was or wasn't received, so it sends three segments starting at segment 4, indicated by PC-B.

Of course, if two segments are lost, the first option listed won't work unless the destination can send a list of lost segments. Therefore, most protocol stacks that use windowing will implement the second option. Given this behavior, the size of the window will really affect your performance. You would normally think that a window size of 100 is the very efficient; however, if the very first packet is lost, some protocols will have *all* 100 packets resent! As mentioned earlier, most protocol stacks use a window size that is negotiated up front and can be re-negotiated at any time. Therefore, if a connection is experiencing a high number of errors, the window size can be dropped to a smaller value to increase your efficiency. And once these errors disappear

exam watch

Ready/not ready signals and windowing are used to implement flow control. Ready/not ready signals are not efficient, causing drops in unnecessary traffic and delays in the transmission of traffic. Windowing addresses these issues. With windowing, a window size is established, which defines the number of segments that can be transferred before waiting for an acknowledgment from the destination.

or drop down to a lower rate, the window size can be increased to maximize your throughput.

What makes this situation even more complicated is that the window sizes on the source and destination devices can be *different*. For instance, PC-A might have a window size of 3, while PC-B has a window size of 10. In this example, PC-A is allowed to send ten segments to PC-B before waiting for an acknowledgment, while PC-B is allowed to send only three segments to PC-A.

Flow control through the use of sequence numbers and acknowledgments is covered in more depth in Chapter 3, where TCP is discussed.

CERTIFICATION OBJECTIVE 2.05

Transferring Information Between Computers

Before delving into the mechanics of how information is transferred between computers, you must grow familiar with the terminology used to describe the transmitted data. Many of the layers of the OSI Reference Model use their own specific terms to describe data transferred back and forth. As this information is passed from higher to lower layers, each layer adds information to the original data—typically a header and possibly a trailer. This process is called *encapsulation*.

Generically speaking, the term *protocol data unit (PDU)* is used to describe data and its overhead. Table 2-13 describes the terms used at the various layers of the OSI Reference Model. For instance, as data is passed from the session layer to the transport layer, the transport layer *encapsulates* the data PDU in a transport layer segment. For TCP and UDP in the TCP/IP protocol stack, the transport layer only adds a header.

TABLE 2-13

PDU Terms

Term	OSI Reference Model Layer
Data	Application, presentation, and session layers
Segment	Transport layer
Packet	Network layer (TCP/IP calls this a *datagram*)
Frame	Data link layer
Bits	Physical layer

As the PDU information is passed down, each layer adds its own header and, possibly, trailer.

Once the physical layer is reached, the bits of the data link layer frame are converted into a physical layer signal—a voltage, light source, radio wave, or other source according to the type of physical medium that is employed. When the destination receives the information, it goes through a reverse process of *de-encapsulating* information—basically stripping off the headers of the PDU information at each layer as the information is passed up from layer to layer of the OSI Reference Model.

Figure 2-10 shows an example of the process used for encapsulating and de-encapsulating PDUs as data is passed down and back up the OSI Reference Model. In this example, you can see how the application, presentation, and session layers create the data PDU. As this information is passed down from layer to layer, each layer adds its own header.

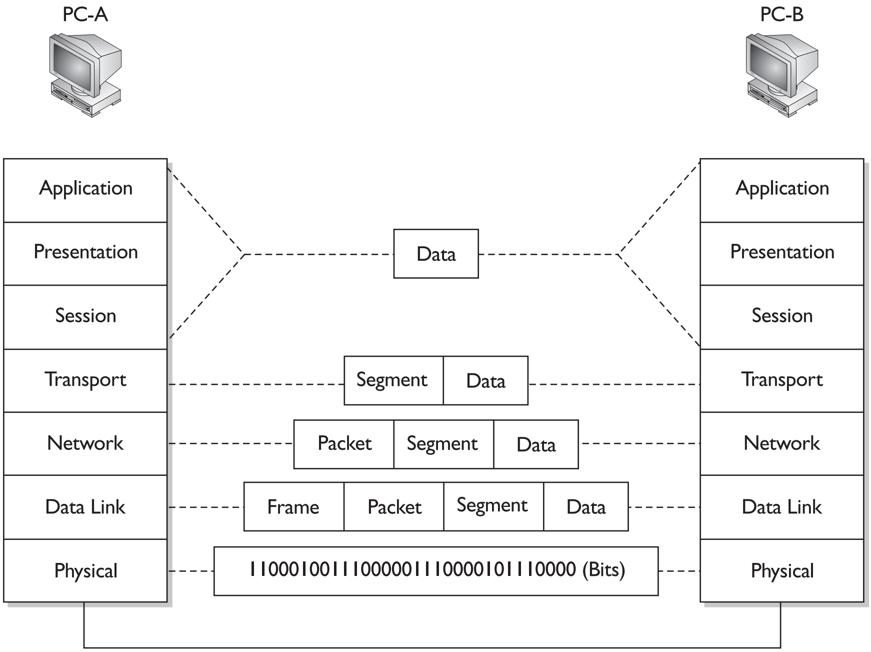
The next few sections are to help you better understand the process that devices go through as information is transmitted between computers. The next section covers the details as to how information is encapsulated and sent down the protocol stack and then placed on the wire to the destination. The section following that covers the reverse process: how the information is de-encapsulated at the destination and delivered to the application at the application layer. The third section looks at a more complex environment, where bridges, routers, and hubs are involved in the communication process to get information from the source to the destination.

Going Down the Protocol Stack

This section covers the basic mechanics as to how information is processed as it is sent down the protocol stack on a computer. I'll use the diagram shown in Figure 2-10 to illustrate this process as PC-A sends information to PC-B. In this example, assume that the data link layer is Ethernet and the physical layer is copper.

FIGURE 2-10

Encapsulation and de-encapsulation process



The first thing that occurs on PC-A is that the user, sitting in front of the computer, creates some type of information, called *data*, and then sends it to another location (PC-B). This includes the actual user input (application layer), as well as any formatting information (presentation layer). The application (or operating system), at the session layer, then determines whether or not the data's intended destination is local to this computer (possibly a disk drive) or a remote location. In this instance, the user is sending the information to PC-B. We'll assume that the user is executing a telnet connection.

The session layer determines that this location is remote and has the transport layer deliver the information. A telnet connection uses TCP/IP and reliable connections (TCP) at the transport layer, which encapsulates the data from the higher layers into a *segment*. With TCP, as you will see in Chapter 3, only a header is added. The segment contains such information as the source and destination port numbers. As you may recall from the section "Connection Multiplexing", the source port is a number above 1,023 that is currently not being used by PC-A. The destination port number is the well-known port number (23) that the destination will understand and forward to the correct application.

The transport layer passes the segment down to the network layer, which encapsulates the segment into a *packet*. The packet header contains layer-3 logical addressing information (source and destination address), as well as other information, such as the upper-layer protocol that created this information. In this example, TCP created this information, so this fact is noted in the packet, and PC-A places its IP address as the source address in the packet and PC-B's as the destination. This helps the destination, at the network layer, to determine if the packet is for itself and which upper-layer process should handle the encapsulated segment. In the TCP/IP protocol stack, the terms *packet* and *datagram* are used interchangeably to describe this PDU. As you will see in Chapter 3, there are many protocols within the TCP/IP protocol stack—TCP, UDP, ICMP, OSPF, and others.

The network layer then passes the packet down to the data link layer. The data link layer encapsulates the packet into a *frame*. If you are using IEEE for the data link layer, remember that two encapsulations take place here: one for LLC and one for MAC. This example uses Ethernet as the data link layer medium, and there are two versions of Ethernet: Ethernet II and IEEE 802.3. To make this more complex, assume the data link layer is based on IEEE's Ethernet implementation. At the LLC sublayer, either an 802.2 SAP or SNAP frame is used. (These frame types were shown previously in Figure 2-4.) TCP/IP uses a SAP frame type. The important information placed in the SAP frame header is which network layer protocol created the packet: IP. The 802.2 SAP frame is then passed down to the MAC sublayer, where the 802.2 frame is encapsulated in an 802.3 frame. The important components placed in the 802.3 frame header are the source and destination MAC addresses. In this example, PC-A places its MAC address in the frame in the source field and PC-B's MAC address as the destination.

The data link layer frame is then passed down to the physical layer. At this point, remember that the concept of "PDUs" is a human concept that we have placed on the data to make it more readable to us, as well as to help deliver the information to the destination. However, from a computer's perspective, the data is just a bunch of 1's and 0's, called *bits*. The physical layer takes these bits and coverts them into a physical property based on the cable or connection type. In this example, the cable is a copper cable, so the physical layer will convert the bits into voltages: one voltage level for a bit value of 1 and a different voltage level for a 0.

Going Up the Protocol Stack

For sake of simplicity, assume PC-A and PC-B are on the same piece of copper. Once the destination receives the physical layer signals, the physical layer translates the voltage levels back to their binary representation and passes these bit values up to the data link layer.

The data link layer takes the bit values and reassembles the original 802.3 frame. The NIC, at the MAC layer, examines the FCS to make sure the frame is valid and examines the destination MAC address to ensure that the Ethernet frame is meant for itself. If the destination MAC address doesn't match its own MAC address, or is not a multicast or broadcast address, the NIC drops the frame. Otherwise, the NIC processes the frame: it strips off the 802.3 frame and passes the 802.2 frame up to the LLC sublayer. The LLC sublayer examines the SAP value to determine which upper-layer protocol at the network layer should process the encapsulated packet. In this case, the LLC sees that the encapsulated packet is a TCP/IP packet, so it strips off (de-encapsulates) the LLC frame information and passes the packet up to the TCP/IP protocol stack at the network layer. If this were an encapsulated IPX packet, the LLC would pass the encapsulated IPX packet up to the IPX protocol stack at the network layer.

The network layer then examines the logical destination address in the packet header. If the destination logical address doesn't match its own address or is not a multicast or broadcast address, the network layer drops the packet. If the logical address matches, then the destination examines the protocol information in the packet header to determine which protocol should handle the packet. In this example, the logical address matches and the protocol is defined as TCP. Therefore, the network layer strips off the packet information and passes the encapsulated segment up to the TCP protocol at the transport layer.

Upon receiving the segment, the transport layer protocol can perform many functions, depending on whether this is a reliable or unreliable connection. I'll just focus on the multiplexing function of the transport layer. In this instance, the transport layer examines the destination port number in the segment header. In our example, the user from PC-A was using telnet to transmit information to PC-B, so the destination port number is 23. The transport layer examines this port number and realizes that the encapsulated data needs to be forwarded to a telnet application. If PC-B doesn't support telnet, the transport layer drops the segment. If it does, the transport layer strips off the segment information and passes the encapsulated data to

exam watch

Be familiar with the process of going down (encapsulation) and back up (de-encapsulation) the protocol stack when transferring data between devices, including when the transfer occurs between devices separated by a switch or a router (described in the next section).

the telnet application. If this is a new connection, a new telnet process is started up by the operating system.

Note that a logical communication takes place between two layers of two devices. For instance, a logical communication occurs at the transport layer between PC-A and PC-B, and this is also true at the network and data link layers.

Two Segment Example

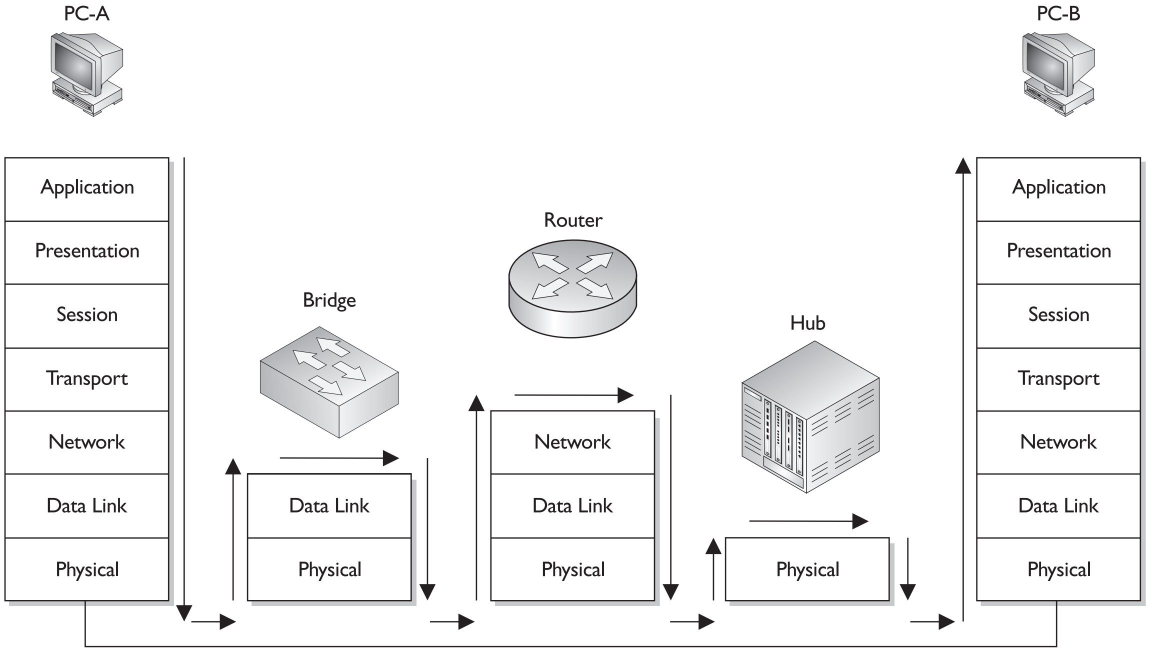
As you can see from the encapsulation and de-encapsulation process, *many* things are occurring on both the source and destination computers to transmit and receive the information. This can become even more complicated if the source and destination are on different segments, separated by other networking devices, such as hubs, bridges, and routers. Figure 2-11 shows an example of this process.

In this example, PC-A wants to send data to PC-B. Notice that each device needs to process information at specific layers. For instance, once PC-A places its information on the wire, the bridge connected to PC-A needs to process this information. Recall from the section "Data Link Layer" of this chapter that bridges function at layer-2 of the OSI Reference Model, making switching decisions based on the destination MAC address found in the frame. Therefore, the bridge's physical layer will have to convert the physical layer signal into bits and pass these bits up to the data link layer, where they are reassembled into a frame. The bridge examines the destination MAC address and makes a switching decision, finding the port the frame needs to exit. It then passes the frame down to the physical layer, where the bits of the frame are converted into physical layer signals.

The next device the physical layers encounter is a router. Recall from the section "Network Layer" of this chapter that routers function at layer-3 of the OSI Reference Model. The router first converts the physical layer signals into bits at the physical layer. The bits are passed up to the data link layer and reassembled into a frame. The router then examines the destination MAC address in the frame. If the MAC address doesn't match its own MAC address, the router drops the frame. If the MAC address matches, the router strips off the data link layer frame and passes the packet up to the network layer.

At the network layer, one of the functions of the router is to route packets to destinations. To accomplish this, the router examines the destination logical address in the packet and extracts a network number from this address. The router then compares the network number to entries in its routing table. If the router doesn't find a match, it drops the packet; if it does find one, the it forwards the packet out the destination interface.

FIGURE 2-11 Multi-segment communications



To accomplish this, the router passes the packet down to the data link layer, which encapsulates the packet into the correct data link layer frame format. If this were an Ethernet frame, for this example, the source MAC address would be that of the router and the destination would be PC-B. At the data link layer, the frame is then passed down to the physical layer, where the bits are converted into physical layer signals.

Note that routers separate physical or logical segments, while bridges (and switches) don't. Therefore, if PC-A wants to send traffic to PC-B, PC-A uses the router's MAC (or layer-2) address to get traffic to the exit point of the segment, but it uses PC-B's logical (or layer-3) address to tell the router that this traffic is not for itself but for a machine on a different segment. This process is discussed in more depth in Chapter 3.

The next device that receives these physical layer signals is the hub. Recall from the section "OSI Reference Model" that hubs and repeaters operate at the physical layer. Basically, a hub is a multiport repeater: it repeats any physical layer signal it receives. Therefore, a signal received on one interface of a hub is repeated on all of its other interfaces. These signals are then received by PC-B, which passes this information up the protocol stack as described in the preceding section.

CERTIFICATION OBJECTIVE 2.06

Hierarchical Network Model

Cisco has developed a three-layer hierarchical model to help you design campus networks. Cisco uses this model to simplify designing, implementing, and managing large-scale networks. With traditional network designs, it was common practice to place the networking services at the center of the network and the users at the periphery. However, many things in networking have changed over the past decade, including advancements in applications, developments in graphical user interfaces (GUIs), the proliferation of multimedia applications, the explosion of the Internet, and fast-paced changes in your users' traffic patterns. Cisco developed the three-layer model to accommodate these rapid changes.

exam watch

The three layers in Cisco's hierarchical design are core, distribution, and access.

Cisco's hierarchical model, shown in Figure 2-12, contains three layers: core, distribution, and access. A well-designed network typically follows this topology. The following sections cover the functions of the three layers, including the devices that function at the various layers.

Core Layer

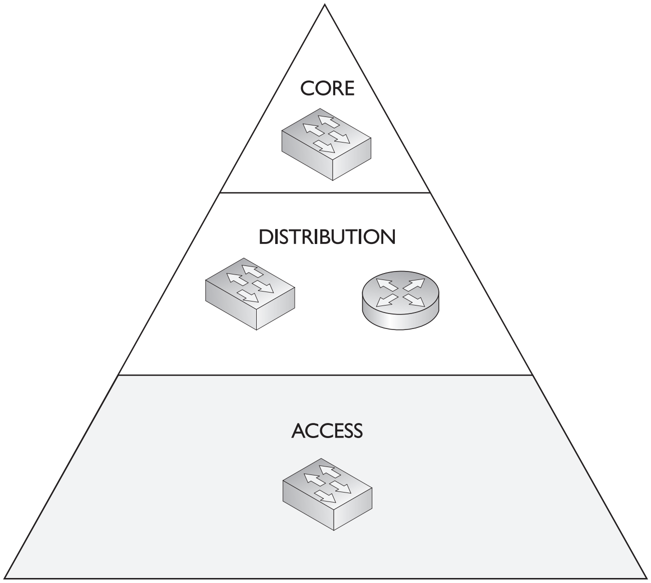
The *core* layer, as its name suggests, is the backbone of the network. It provides a high speed connection between the different distribution layer devices. Because of the need for high-speed connections, the core consists of high-speed *switches* and will not, typically, perform any type of packet or frame manipulations, such as filtering or Quality of Service. Because switches are used at the core, the core is referred to as a layer-2 core. The traffic that traverses the core is typically to access enterprise corporate resources: connections to the Internet, gateways, e-mail servers, and corporate applications.

exam watch

The core provides a high-speed layer-2 switching infrastructure and typically does not manipulate packet contents.

FIGURE 2-12

Cisco's hierarchical model



Distribution Layer

Of the three layers, the *distribution* layer performs most of the connectivity tasks. In larger networks, routers are used at the distribution layer to connect the access layers to the core. For smaller networks, sometimes switches are used. The responsibilities of the distribution layer include the following:

- Containing broadcasts between the layers
- Securing traffic between the layers
- Providing a hierarchy through layer-3 logical addressing and route summarization
- Translating between different media types

As mentioned in the section "Network Layer", routers give you by far the most flexibility in enforcing your company's networking policies, since routers deal with logical addresses. And because routers are used at the distribution layer, the implementation of your policies, at least most of them, is done here.

Containing Broadcasts

One of the main functions of the distribution layer is to contain broadcast and multicast traffic that the access layer devices create. If a broadcast storm is created in one access layer, or there is a large amount of multicast traffic from a real-time video stream, the distribution layer, by default, confines this traffic in the access layer and thus prevents it from creating problems in other areas.

Providing Logical Addressing

Routers also provide for logical addressing of devices in your network. This makes it much easier to implement your networking policies, including filtering and QoS since you control how addresses are assigned to machines: it is very difficult to do this with layer-2 MAC addresses. Another advantage that logical addressing provides is that, again, with the correct address layout in your network, you should be able to create a highly scalable, hierarchical network. This topic is discussed in Chapter 12.

Performing Security

Another function of this layer is to enforce your security policies. Because switches are used at the core and access layers, security is not typically implemented at these layers, given the issues of filtering MAC addresses. Since routers deal with logical addresses, however, they make it much easier to implement your policies. In Chapter 13, you'll see how you can use access lists to implement your security policies.

Connecting Different Media Types

If you have two different media types that you want to connect, Token Ring and Ethernet, for instance, a router is the best solution; and since routers are used at the distribution layer, this is where this conversion takes place. As mentioned in the section "Data Link Layer", bridges are not very good at performing translations between different media types. However, routers do not have this problem, as you saw in the section *Two Segment Example*. Routers don't translate between media types. Instead, they perform a de-encapsulation and encapsulation process. From layer-2, the router strips off the frame and passes up the packet to layer-3. At layer-3, the router makes its routing decision and queues the packet on the outbound interface. Once again, at layer-2, the packet is encapsulated in the frame type for the corresponding media type the interface is connected to.

exam watch

The distribution layer provides a boundary between the access and core layers. It contains routers and switches. Routers are used to provide the logical boundary--broadcasts are contained within the access layer and filtering policies can be implemented to restrict traffic flows.

Access Layer

The bottom layer of the three-layer hierarchical model is the *access* layer. Actually, the access layer is at the periphery of your campus network, separated from the core layer by the distribution layer. The main function of the access layer is to provide the user's initial connection to your network. Typically, this connection is provided by a switch, or sometimes, a hub. Sometimes if the user works at a small branch office or home office, this device can also be a router. But in most cases, the connection is provided by a switch.

exam watch

The access layer provides the user's initial access to the network, which is typically via switches or hubs.

Connections

Remember that the three-layer hierarchical model is a logical, not a physical, representation. For example, sometimes the distribution layer device might contain both switches and routers. You might, say, use a Catalyst 5000 switch with a route switch module (RSM) in its chassis—this combination of devices can provide both layer-2 and layer-3 functionality at the distribution layer. This kind of setup is common at the distribution layer: sometimes the routing function sits inside the chassis of the switch, like the MSM card for a Catalyst 6000 switch, and sometimes the routing function is in a separate chassis, like a 3600 series router. No matter what configuration is used, it is important that you configure the layer-3 device correctly to create a boundary between the access and core layer devices. The switching function that can be provided by the distribution layer is used to connect departmental services that the access layer devices commonly access.

Remember that, since this is a hierarchical model, connections should always be made in the upward direction: access-to-distribution and distribution-to-core. You should never cross-connect layers: access-to-access or distribution-to-distribution. If you do this, you'll be creating a non-scalable flat network. Cisco's CCDA, BCMSN, and ARCH exams cover this process in more depth.

CERTIFICATION SUMMARY

The OSI Reference Model defines the process of connecting two layers of networking functions. The application layer provides the user's interface. The presentation layer determines how data is represented to the user. The session layer is responsible for setting up and tearing down connections. The transport layer is responsible for the mechanics of connections, including guaranteed services. The network layer provides a logical topology and layer-3 addresses: routers operate here. The data link layer defines MAC addresses and how communication is performed on a specific media type: switches, bridges, and NICs operate here. The physical layer defines physical properties for connections and communication: repeaters and hubs operate here. Wireless solutions are defined at the physical layer. The 802.11 services define wireless access. Wi-Fi is defined by 802.11b.

The data link layer defines hardware addressing. MAC addresses are 48-bits in length in hexadecimal. The first 24 bits (6 digits) are the OUI. MAC addresses only need to be unique on a logical segment. A unicast is where one frame is sent to a single device. A multicast is where one frame is sent to a group of devices. A broadcast is where one frame is sent to all devices.

CSMA/CD is used to implement Ethernet. Ethernet is a shared medium. When a device wants to transmit, it must first listen to the wire to determine if a transmission is already occurring. If two devices try to simultaneously send their transmissions, a collision occurs. When this happens, a jam signal is created and the two devices back off a random period before trying again. There are two versions of Ethernet: IEEE 802.2/3 and Ethernet II (or DIX). 802.2 defines the LLC (software) and 802.3 defines the MAC (hardware). 802.2 uses a SAP or SNAP field to designate the layer-3 encapsulated protocol. Ethernet II doesn't have any sublayers and doesn't have a length field; instead, it has a type field.

Half-duplex connections allow a device to either send or receive at a time while full-duplex allow simultaneous sending and receiving. Full-duplex require a point-to-point connection.

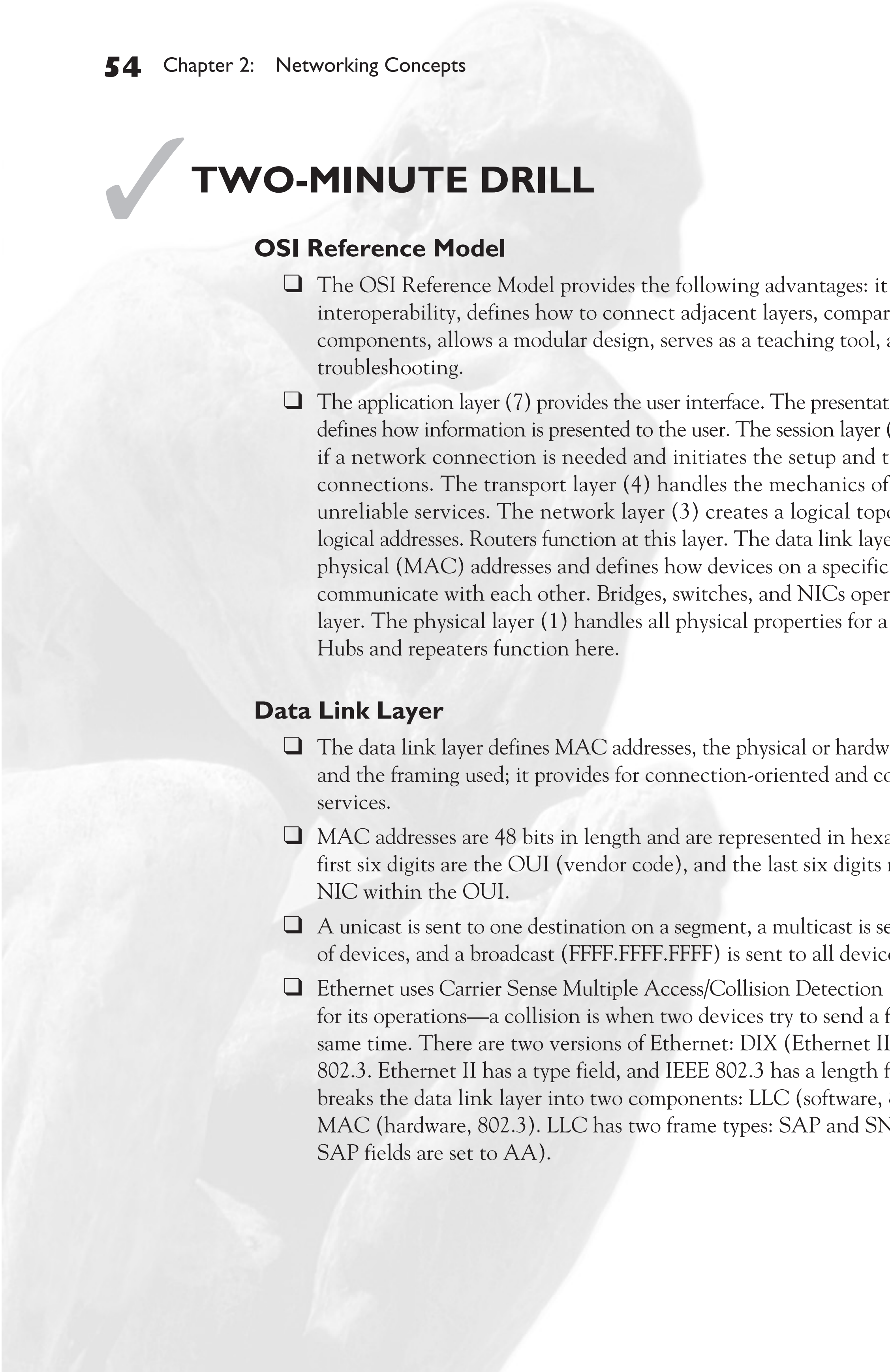
The three main functions of a bridge and switch is to learn where devices are located, forward traffic using their port address table, and remove loops using STP. Unknown unicast addresses, broadcasts, and multicasts are flooded by a bridge and switch. Bridges and switches are used to solve bandwidth and collision problems.

The network layer defines logical addresses, finds paths to destinations based on the network component of the address, and connect different layer-2 media types together. Routers are used to contain broadcasts. Routers use their routing table, which has a list of destination network numbers, to assist them when finding a destination. If a destination is not found in the routing table, traffic for this destination is dropped.

The transport layer sets up and maintains a session layer connection, provides for reliable or unreliable delivery of data, flow control, and multiplexing of connections. Reliable connections go through a three-way handshake to establish a connection: SYN, SYN/ACK, and ACK. Acknowledgements are used to provide reliable delivery. Port or socket numbers are used for connection multiplexing. Ready/not ready signals and windowing are used to implement flow control. Windowing is more efficient than ready/not ready signals.

A PDU describes data and its overhead. A PDU at the application layer is referred to as data; the transport layer PDU is called a segment, the network layer PDU is called a packet or datagram, the data link layer PDU is called a frame and the physical layer PDU is called bits. As traffic goes down the protocol stack, each layer encapsulates the PDU from the layer above it. At the destination, a de-encapsulation process occurs.

Cisco uses a hierarchical model to help with designing networks. The core layer provides a high-speed layer-2 infrastructure and typically doesn't manipulate packet contents. The distribution layer separates the access and core layers, typically through a layer-3 process. It uses routers and switches. The access layer provides a user's initial connection to the network, typically done by a switch.



TWO-MINUTE DRILL

OSI Reference Model

□ The OSI Reference Model provides the following advantages: it promotes interoperability, defines how to connect adjacent layers, compartmentalizes components, allows a modular design, serves as a teaching tool, and simplifies troubleshooting.

□ The application layer (7) provides the user interface. The presentation layer (6) defines how information is presented to the user. The session layer (5) determines if a network connection is needed and initiates the setup and teardown of connections. The transport layer (4) handles the mechanics of reliable or unreliable services. The network layer (3) creates a logical topology with logical addresses. Routers function at this layer. The data link layer (2) assigns physical (MAC) addresses and defines how devices on a specific media type communicate with each other. Bridges, switches, and NICs operate at this layer. The physical layer (1) handles all physical properties for a connection. Hubs and repeaters function here.

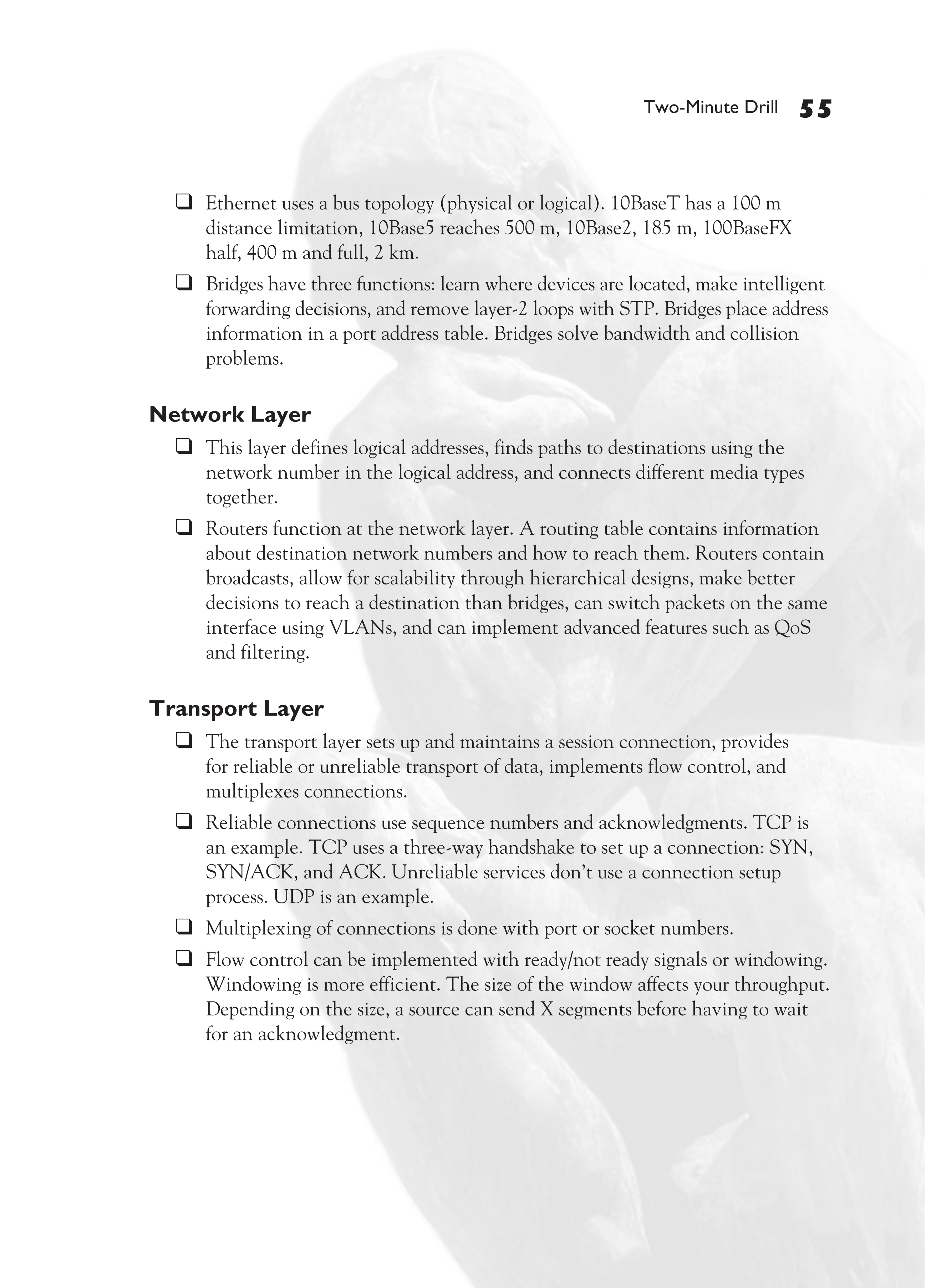
Data Link Layer

□ The data link layer defines MAC addresses, the physical or hardware topology, and the framing used; it provides for connection-oriented and connectionless services.

□ MAC addresses are 48 bits in length and are represented in hexadecimal. The first six digits are the OUI (vendor code), and the last six digits represent the NIC within the OUI.

□ A unicast is sent to one destination on a segment, a multicast is sent to a group of devices, and a broadcast (FFFF.FFFF.FFFF) is sent to all devices.

□ Ethernet uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD) for its operations—a collision is when two devices try to send a frame at the same time. There are two versions of Ethernet: DIX (Ethernet II) and IEEE 802.3. Ethernet II has a type field, and IEEE 802.3 has a length field. IEEE breaks the data link layer into two components: LLC (software, 802.2) and MAC (hardware, 802.3). LLC has two frame types: SAP and SNAP (the SAP fields are set to AA).



❑ Ethernet uses a bus topology (physical or logical). 10BaseT has a 100 m distance limitation, 10Base5 reaches 500 m, 10Base2, 185 m, 100BaseFX half, 400 m and full, 2 km.

❑ Bridges have three functions: learn where devices are located, make intelligent forwarding decisions, and remove layer-2 loops with STP. Bridges place address information in a port address table. Bridges solve bandwidth and collision problems.

Network Layer

❑ This layer defines logical addresses, finds paths to destinations using the network number in the logical address, and connects different media types together.

❑ Routers function at the network layer. A routing table contains information about destination network numbers and how to reach them. Routers contain broadcasts, allow for scalability through hierarchical designs, make better decisions to reach a destination than bridges, can switch packets on the same interface using VLANs, and can implement advanced features such as QoS and filtering.

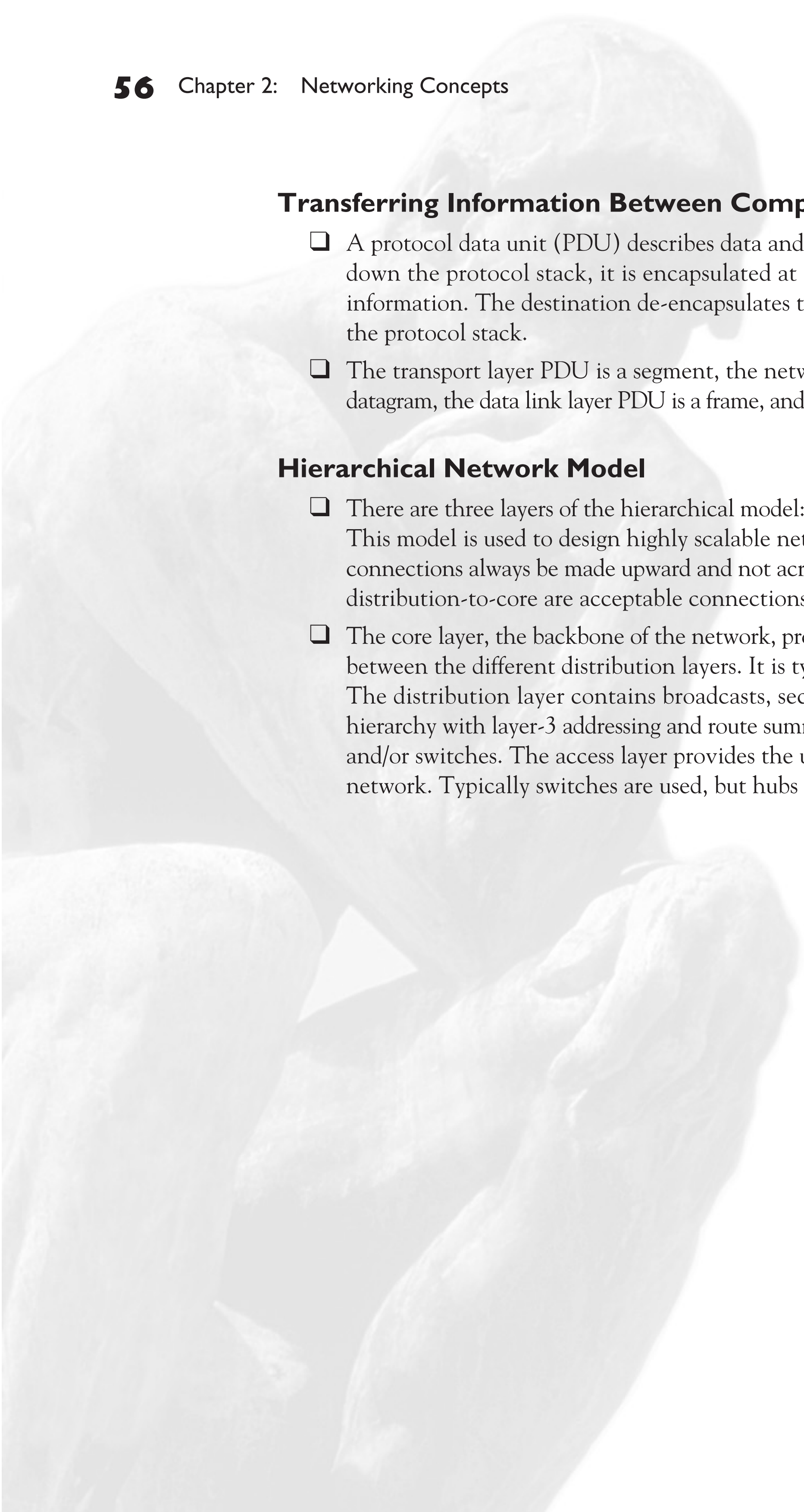
Transport Layer

❑ The transport layer sets up and maintains a session connection, provides for reliable or unreliable transport of data, implements flow control, and multiplexes connections.

❑ Reliable connections use sequence numbers and acknowledgments. TCP is an example. TCP uses a three-way handshake to set up a connection: SYN, SYN/ACK, and ACK. Unreliable services don't use a connection setup process. UDP is an example.

❑ Multiplexing of connections is done with port or socket numbers.

❑ Flow control can be implemented with ready/not ready signals or windowing. Windowing is more efficient. The size of the window affects your throughput. Depending on the size, a source can send X segments before having to wait for an acknowledgment.



Transferring Information Between Computers

□ A protocol data unit (PDU) describes data and its overhead. As data is sent down the protocol stack, it is encapsulated at each layer by additional information. The destination de-encapsulates the data as it goes back up the protocol stack.

□ The transport layer PDU is a segment, the network layer PDU is a packet or datagram, the data link layer PDU is a frame, and the physical layer PDU is bits.

Hierarchical Network Model

□ There are three layers of the hierarchical model: core, distribution, and access. This model is used to design highly scalable networks. It is recommended that connections always be made upward and not across: access-to-distribution and distribution-to-core are acceptable connections.

□ The core layer, the backbone of the network, provides high-speed connections between the different distribution layers. It is typically composed of switches. The distribution layer contains broadcasts, secures traffic, and provides a hierarchy with layer-3 addressing and route summarization. It consists of routers and/or switches. The access layer provides the user's initial connection to the network. Typically switches are used, but hubs and routers can also be used.

SELF TEST

The following Self Test questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question.

OSI Reference Model

1. Put the following in the correct order, from high to low: session (a), presentation (b), physical (c), data link (d), network (e), application (f), transport (g).

A. c, d, e, g, a, b, f
B. f, a, b, g, d, e, c
C. f, b, g, a, e, d, c
D. f, b, a, g, e, d, c

2. Which of the following standards or protocols are used by the session layer?

A. JPEG
B. NFS
C. TCP
D. Ethernet

3. The __________ layer provides for hardware addressing.

A. Transport
B. Network
C. Data link
D. Physical

Data Link Layer

4. MAC addresses are __________ bits in length and are represented in a ____________ format.

5. CSMA/CD stands for _________.

A. Collision Sense Multiple Access/Carrier Detection
B. Carrier Sense Multiple Access/Collision Detection
C. Collision Sense Media Access/Carrier Detection
D. Carrier Sense Media Access/Collision Detection

6. Ethernet ________ uses a type field.

A. II

B. 802.3

C. 802.2

7. Which component of the data link layer for IEEE specifies network protocols?

A. LLC

B. MAC

C. 802.5

D. 802.3

Network Layer

8. The network layer solves all of the following problems except ___________.

A. Broadcast problems

B. Conversion between media types

C. Hierarchy through the use of physical addresses

D. Collision problems

9. If a router has a packet it needs to route, and it can't find the destination network number in the routing table, the router ________ the packet.

A. Drops

B. Floods

Transport Layer

10. ________ are used to provide a reliable connection.

A. Ready/not ready signals

B. Sequence numbers and acknowledgments

C. Windows

D. Ready/not ready signals and windowing

11. Connection multiplexing is done through the use of a ________ number.

A. Socket

B. Hardware

C. Network

D. Session

12. Reliable connections go through a three-way handshake. Place the following in the correct order: ACK (1), SYN, (2), SYN/ACK (3).

A. 2, 1, 3
B. 3, 2, 1
C. 2, 3, 1
D. 1, 2, 3

Transferring Information Between Computers

13. The network layer transmits _________ PDUs.

A. Datagram
B. Segment
C. Bits
D. Frame

Hierarchical Network Design

14. The ________ layer uses high-speed switches to provide high-speed connections.

A. Core
B. Access
C. Distribution

15. The ________ layer contains broadcasts.

A. Core
B. Access
C. Distribution

SELF TEST ANSWERS

OSI Reference Model

1. ☑ **D.** From high to low, the OSI Reference Model has the following layers: application, presentation, session, transport, network, data link, and physical.
☒ **A** doesn't begin with the application layer. **B**'s second from the top layer is presentation. **D** switches the session and transport layers.

2. ☑ **B.** NFS is a session layer protocol.
☒ **A** is a presentation layer standard. **C** is a transport layer protocol. **D** is a data link layer standard.

3. ☑ **C.** The data link layer provides for hardware addressing.
☒ **A** uses port numbers for multiplexing. **B** defines logical addressing. **D** doesn't have any addressing.

Data Link Layer

4. MAC addresses are 48 bits in length and are represented in a hexadecimal format.

5. ☑ **B.** CSMA/CD stands for Carrier Sense Multiple Access/Collision Detection. It defines how Ethernet functions.

6. ☑ **A.** Ethernet II uses a type field.
☒ **B** uses a length field. **C** is not an Ethernet standard.

7. ☑ **A.** The LLC uses SAPs to define network layer protocols.
☒ **B** defines hardware addresses. **C** is Token Ring, and **D** is Ethernet.

Network Layer

8. ☑ **C.** The network layer creates a hierarchy through the use of logical, not physical addresses.
☒ **A**, **B**, and **D** are true and thus incorrect.

9. ☑ **A**. If a router has a packet it needs to route, and it can't find the destination network number in the routing table, the router *drops* the packet.
☒ **B** is true of switches, not routers.

Transport Layer

10. ☑ **B.** Sequence numbers and acknowledgments are used to provide a reliable transport layer connection.
☒ **A**, **C**, and **D** are used for flow control.

11. ☑ **A.** Connection multiplexing is done through the use of a socket or port number.
☒ **B** references the data link layer. **C** references the network layer. **D** is a nonexistent number type.

12. ☑ **C.** Reliable transport layer connections go through a three-way handshake process: SYN, SYN/ACK, ACK.
☒ **A**, **B**, and **D** are incorrect because they specify the wrong order.

Transferring Information Between Computers

13. ☑ **A.** The network layer transmits packet or datagram PDUs.
☒ **B** is true for the session layer. **C** is true for the physical layer. **D** is true for the data link layer.

Hierarchical Network Design

14. ☑ **A.** The core layer uses high-speed switches to provide high-speed connections.
☒ **B** provides the users' initial connection to the network. **C** provides a boundary between access and core and allows you to implement your policies.

15. ☑ **C.** The distribution layer contains broadcasts and multicasts.
☒ **B** provides the users' initial connection to the network. **A** provides a high-speed switching infrastructure to connect different distribution layers.