Microsoft

CompTIA® Network+®

Exam N10-005

FREE SAMPLER

Craig Zacker

Training Kit

Contents

CHAPTER 1

Networking Basics

This chapter introduces the fundamental computer networking concepts that form the basis for all of the questions on the CompTIA Network+ examination. You might be inclined to skip around in this book during your exam preparation regimen, but you should make sure that you understand the principles in this chapter before you do so. Otherwise, you might find yourself struggling later, both in the exam room and on the job.

IMPORTANT

Have you read page xxii?

It contains valuable information regarding the skills you need to pass the exam.

Exam objectives in this chapter:

Objective 1.1: Compare the layers of the OSI and TCP/IP models.

- OSI model:
  - Layer 1 – Physical
  - Layer 2 – Data link
  - Layer 3 – Network
  - Layer 4 – Transport
  - Layer 5 – Session
  - Layer 6 – Presentation
  - Layer 7 – Application
- TCP/IP model:
  - Network Interface Layer
  - Internet Layer
  - Transport Layer
  - Application Layer
    - (Also described as: Link Layer, Internet Layer, Transport Layer, Application Layer)

Objective 1.2: Classify how applications, devices, and protocols relate to the OSI model layers.

- MAC address
- IP address
- EUI-64
- Frames
- Packets
- Switch
- Router
- Multilayer switch
- Hub
- Encryption devices
- Cable
- NIC
- Bridge

REAL WORLD **REINVENTING NETWORK+**

Anyone familiar with the earlier incarnations of the CompTIA Network+ examination might notice that there are some rather profound differences between the objectives tested by the N10-004 version of 2009 and those in the N10-005 version released in late 2011. Some of these changes are representative of the latest developments in networking technology, and others demonstrate a definite change in the focus of the exam.

First, and most obvious, is the elimination of many technologies that have lapsed into obsolescence. With Ethernet now unquestionably the dominant data-link layer protocol on the desktop, older protocols such as Token Ring and Fiber Distributed Data Interface (FDDI), which were included in the 2005 edition of the objectives, are now gone. Conversely, the 802.11 wireless LAN standards that barely rated a mention in 2005 and received two objectives in 2009 now have four, making them a major part of the exam.

At the network and transport layers, TCP/IP is now ubiquitous, displacing older alternatives such as IPX/SPX, NetBEUI, and AppleTalk. This is not to say that you will never encounter any of these protocols in the field ever again, but they are now considered rare, if not actually endangered, species.

Whereas the 2005 objectives specified the need for basic knowledge of various server operating systems, the 2009 and 2011 objectives place far more concentration on specific areas of network support, such as configuration management, performance optimization, and troubleshooting methodologies. The operating system names no longer appear in the objectives at all.

The 2011 objectives also clarify the examination's emphasis on infrastructure management. New objectives single out services such as DNS and DHCP for particular concentration and deemphasize hardware and software technologies that are fading from general use.

Network Communications

What is a data network? Simply put, a data network is an array of computers and other devices connected together by a common medium that enables them to communicate with each other. That common medium can be wired, using copper or fiber optic cables; wireless, using infrared or radio signals; or connected to a service provider, such as a telephone or cable television network. A data network can be as simple as two home computers connected together, or as complex as the Internet, joining millions of computers together around the world.

Why connect computers together? The two primary reasons to create data networks are to:

- Share hardware
- Share data

In the early days of the PC, the only way to print a document was to connect a printer directly to a computer. As more and more companies adopted the PC as a business tool, it became impractical to buy a printer for every computer or to move a single printer from computer to computer as needed. By connecting computers to a network, they could share a single printer.

In the same way, networking made it possible for computers to share data. Rather than save a document file to a removable disk and walk it to another computer—a process colloquially known as the *sneakernet*—users could store files on a common server, enabling anyone to access them. As networks grew larger and more complex, so did the applications that made use of them. Today, in addition to document files and printer jobs, networks carry data in the form of email messages, webpages, video streams, and many other types.

LANs and WANs

The earliest PC networks used copper-based cables as the network medium, and many still do. A *local area network (LAN)* is a group of computers or other devices that share a common location, such as a room, a floor, or a building; and a common network medium, such as a particular type of cable. The medium interconnects the computers so that they are capable of sharing data with each other. LANs can include network connection devices, such as switches and routers, and are also characterized by their relatively high data transmission rates and their ability to function without the need for outside service providers.

A typical small LAN is shown in Figure 1-1. LANs are wholly owned by an organization and require no licensing or registration. Anyone can purchase the hardware required and assemble a LAN in his or her home or office.

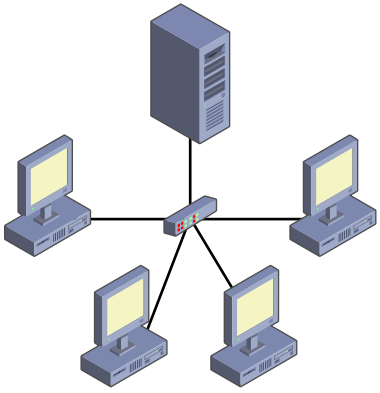

FIGURE 1-1 A typical small LAN.

Devices connected to a LAN, such as computers or printers, are generically referred to as *nodes*. A 50-node network is therefore a single network medium with 50 computers or other devices connected to it.

EXAM TIP

Virtually all of the wired LANs installed today use a technology known as Ethernet or, more precisely, IEEE 802.3. There are several other antiquated LAN technologies, including Token Ring and FDDI, that are no longer covered by the Network+ exam, and for which products are no longer available on the market, but that you might conceivably encounter in older installations.

LANs are expandable within certain limits imposed by the protocols they use to communicate, but in large installations, it is often necessary to connect multiple LANs together. To do this, you use a device called a router, as shown in Figure 1-2. A *router* is simply a device that connects networks together, forming what is known as a "network of networks" or, more commonly, an "internetwork."

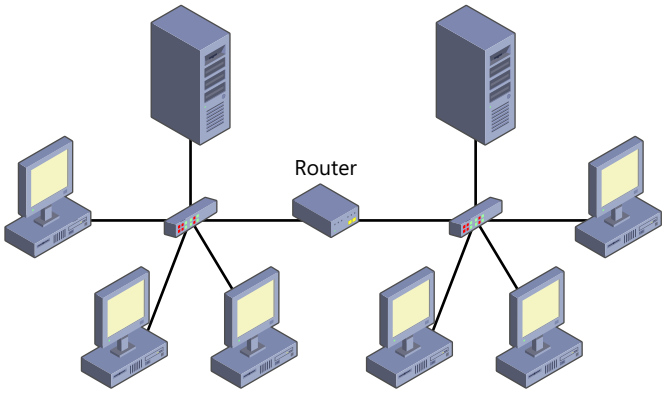

FIGURE 1-2 Two LANs connected by a router.

NOTE **Internet or internet?**

Do not confuse the terms "internetwork" or "internet" (with a lowercase "i") with the Internet (with a capital "I"). The term "Internet" describes a specific example of that for which "internetwork" is the generic designation. In other words, the Internet is a specific type of internetwork, but not every use of the term "internetwork" refers to the Internet.

A *wide area network (WAN)* is a group of computers connected by a longer distance communication technology provided by a third-party service provider, such as a telephone company. Internet connections for LANs or individual computers, whether they use dial-up modems and telephone lines or broadband technologies, are all WAN links. Corporate networks also use WAN technologies to connect offices at remote sites together. Most WAN connections are point-to-point links joining two sites together; a company with multiple branch offices in different cities might have separate WAN links connecting each branch to the main office. As with LANs, WANs are connected together by routers, as shown in Figure 1-3.

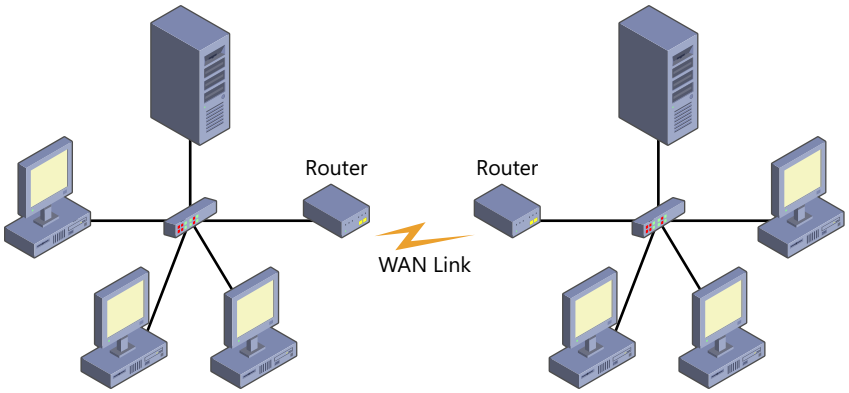

FIGURE 1-3 Two LANs connected by a WAN link.

MORE INFO **WAN TECHNOLOGIES**

For more information on the various types of WAN technologies currently in use, see Chapter 10, "Wide Area Networking."

WAN connections can take many forms and use many different communications technologies. Subscribers, whether private individuals or large companies, can choose from among a variety of WAN providers offering connections with different speeds and services. Generally speaking, WAN connections are much slower than LAN connections and are far more expensive. Most LANs today run at 100 or 1,000 megabits per second (Mbps), and the only costs involved are for the required hardware components. WAN connections typically run at speeds of up to 4 Mbps for residential Internet connections, and up to 25 Mbps for business connections. Very few even approach the speed of a modest LAN. Subscription prices vary depending on the speed of the connection and the other services provided.

Signals and Protocols

All of the computers connected to a network communicate by exchanging signals with each other. The nature of the signals depends on the network medium. The three most common types of signals used for network communications are as follows:

- **Electrical** Networks that use copper-based cables as a medium communicate by using electrical signals, voltages generated by the transceiver in each node.
- **Light** Fiber optic cables carry signals in the form of pulses of light, and some wireless networks use infrared light as a signaling medium.
- **Radio** Most wireless networks communicate by using radio signals.

In each of these cases, the signals form a simple code that enables the computers to transmit data over the network. At the signaling level, network communications are extremely simple, consisting only of positive or negative voltages, the presence or absence of light, or

specific radio frequency variations. The process by which complex data structures, such as print jobs, email messages, and video streams, get reduced to simple signals is the responsibility of software components called protocols, which run on each computer.

Protocols are essentially languages that operate at various levels of the networking software on each computer or other device. Just as two people must speak the same language to be able to talk to each other, two computers on the same network must use the same protocols to communicate. Unlike human speech, however, which uses a single language, a networked computer uses multiple protocols in layers, forming a construction known as a *protocol stack*.

The signals that the computer transmits over the network medium are at the bottom of the stack, and the applications that handle the data are at the top, as shown in Figure 1-4. One of the primary functions of the protocol stack is to reduce the data generated by the applications running on the computer down to the simple signals suitable for the network medium. When the signals arrive at their destination, the protocol stack performs the same process in reverse, interpreting the incoming signals and restoring them to their original form.

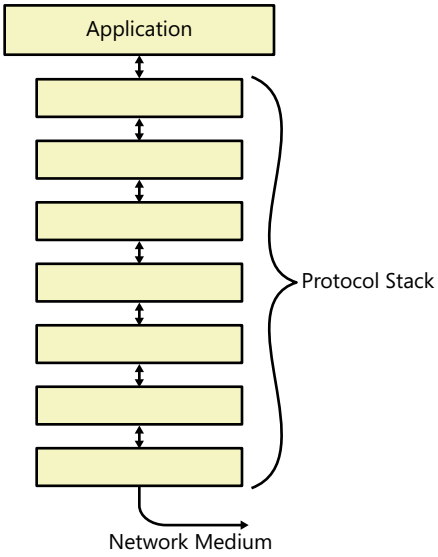

FIGURE 1-4 The protocol stack on a networked computer.

Ethernet, TCP, IP, and SMTP are all protocols operating at various layers of a typical networked computer's protocol stack. A large part of the Network+ exam is devoted to testing your knowledge and understanding of these various protocols.

Networking protocols can provide many different functions to the data structures on which they operate. The protocols protect the data as the computers transmit it over the network and see to it that the data arrives at its intended destination. Some of the most important protocol functions are described in the following list.

- **Addressing** A system for assigning a unique designation to each computer on a network and using those designations to transmit data to specific computers
- **Acknowledgment** The transmission of a return message by the receiving system to verify the receipt of data
- **Segmentation** The division of a large block of data into segments sufficiently small for transmission over the network
- **Flow control** The generation of messages by a receiving system that instruct the sending system to speed up or slow down its rate of transmission
- **Error detection** The inclusion of special codes in a data transmission that the receiving system uses to verify that the data was not damaged in transit
- **Error correction** The retransmission of data that has been corrupted or lost on the way to its destination
- **Encryption** The encoding of data with a cryptographic key to protect it during transmission over the network
- **Compression** The removal of redundant information from data blocks to minimize the amount of data transmitted over the network

Most of the networking protocols in use today are based on open standards so that different manufacturers can produce implementations that are fully compatible. Some of the organizations that are responsible for designing the networking protocol standards are as follows:

- **Institute of Electrical and Electronics Engineers (IEEE)** The US-based society responsible for the publication of the IEEE 802.3 working group, which includes the standards that define the protocol commonly known as Ethernet, as well as many others.
- **International Organization for Standardization (ISO)** A worldwide federation of standards bodies from more than 100 countries, responsible for the publication of the Open Systems Interconnection (OSI) reference model document.
- **Internet Engineering Task Force (IETF)** An ad hoc group of contributors and consultants that collaborates to develop and publish standards for Internet technologies, including the Transmission Control Protocol/Internet Protocol (TCP/IP) protocols.
- **American National Standards Institute (ANSI)** A private, nonprofit organization that administers and coordinates the United States' voluntary standardization and conformity assessment system. ANSI is the official US representative to the ISO, as well as to several other international bodies.
- **Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA)** Two organizations that have joined together to develop and publish the Commercial Building Telecommunications Wiring Standards, which define how the cables for data networks should be installed.
- **Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T)** An international organization within which governments and the private sector work together to coordinate the operation of telecommunication networks and services, and to advance the development of communications technology.

Packet Switching and Circuit Switching

LANs typically use a shared network medium, meaning that all of the computers are connected to a medium that can only carry one signal, and they must take turns using it. A network that can only carry one signal at a time is called a *baseband network*.

When multiple computers have to share a single baseband network medium, they must transmit their data in the form of small, discrete units called *packets*. Otherwise, one computer might monopolize the network for long periods of time as it transmits large files. Instead of transmitting an entire file all at once, the protocols running on the computer break it down into packets and transmit them to the destination individually. This way, many computers can gain access to the network and take turns transmitting packets. If you were to imagine the network cable as a hose that has been cut with a knife, you would see packets originating from many different computers squirt out onto the floor, as shown in Figure 1-5.

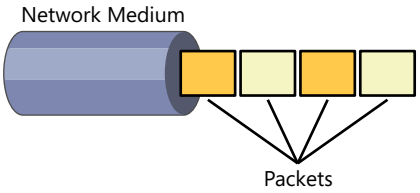

FIGURE 1-5 Packets transmitted over a baseband network.

Because computers on a baseband network have to break up their transmissions into separate packets, it is conceivable that the packets that compose a single file might take different routes to their destination, and might even arrive at the destination out of order. The destination system must therefore identify the incoming packets and reassemble them in the proper order to recreate the original data forms transmitted by the sender. This type of arrangement is known as a *packet-switching* network.

The opposite of a packet-switching network is a *circuit-switching* network, in which one system opens a circuit (or path) to another system prior to transmitting any data. The circuit then remains open for the duration of the data transaction.

Circuit-switching is not suitable for LANs, because it would monopolize the network medium for long periods. An example of circuit-switching is the Public Switched Telephone Network (PSTN) through which you receive all of your land-line telephone calls. When you pick up the ringing phone, it establishes a circuit to the caller's phone, and that circuit remains open—even when nobody is talking—until one of you hangs up the receiver.

To make circuit switching practical, some networks use an alternative to baseband communications called *broadband*. In recent years, the term "broadband" has come to refer to any high-speed Internet connection, but the actual definition of a broadband network is one that can carry multiple signals on a single network medium. Broadband networks use a technique called *multiplexing* to divide the bandwidth into separate channels, each of which can carry a different signal.

One of the most common types of broadband networks is the cable television (CATV) network connection found in many homes. The CATV service enters the home as a single cable, but if you have multiple television sets, each one can be tuned to a different channel. This means that the single cable is carrying the signals for dozens (or hundreds) of different channels simultaneously. That same cable can supply the home with other services as well, including Internet access and video on demand.

Client/Server and Peer-to-Peer Networks

The basic functions of a network typically involve one computer or other device providing some kind of service to other computers. This relationship is typically referred to as *client/server networking*. The server side of the partnership can be a computer that provides storage, access to a printer, email services, webpages, or any number of other services. The client is a computer running a program that accesses the services provided by servers.

In the early days of PC networking, these client and server roles were more clearly defined than they are today. Servers were computers dedicated exclusively to server functions; they could not function as clients.

Today, virtually all of the computers on a network are capable of functioning as both clients and servers simultaneously, and their roles are more a matter of the administrator's choice than the software running on the computer. This relationship is known as *peer-to-peer networking*. On a peer-to-peer network, for example, a computer can share its drives with the rest of the network and can also access shared drives on other computers, regardless of whether the system is running a server or a client operating system.

NOTE **CLIENT/SERVER OPERATING SYSTEMS**

One of the few successful network operating system products that operated on a strictly client/server model was Novell NetWare. All Windows, UNIX, and Linux operating systems are capable of using the peer-to-peer model.

Manufacturers of operating systems still tend to market separate server and client versions of their products, but a computer running a server operating system can still function as a client, and many home or small business networks consist solely of computers running client operating systems, which can also function as servers at the same time.

This might seem confusing, but suffice it to say that in today's computing world, the terms "client" and "server" refer not so much to machines or operating systems as they do to the roles or applications running on those machines or operating systems.

The OSI Reference Model

As mentioned earlier in this chapter, networked computers use protocols to communicate with each other, and the combination of protocols running at the same time in a network implementation is called the stack. The *Open Systems Interconnection (OSI) reference model* is a theoretical example of a network protocol stack, which networking students and administrators use to categorize and define a computer's various networking functions.

The OSI reference model consists of seven layers, which are as follows, from top to bottom:

- 7 - Application
- 6 - Presentation
- 5 - Session
- 4 - Transport
- 3 - Network
- 2 - Data-link
- 1 - Physical

> ***NOTE*** **OSI LAYER NUMBERS**
>
> **The upper layers of the OSI model are seldom referenced by number. The most common use for the layer numbers is in discussions of routing and switching technologies. Switches operate primarily at Layer 2, the data-link layer, and routers at Layer 3, the network layer. However, these devices often have capabilities that span to other layers, resulting in references to technologies such as Layer 3 switching. For more information, see Chapter 7, "Routing and Switching."**

The top of the model interacts with the applications running on the computer, which might at times require the services of the network. The bottom of the model connects to the network medium over which the system transmits its signals, as shown in Figure 1-6. There are different protocols operating at the various layers of the model, each of which provides functions needed to compete the network communication process.

When an application on a computer requests a network service, such as access to a file on a server or the transmission of an email message, it generates a network access request and passes it to a protocol operating at the application layer. For example, to access a file on a server, a Windows-based computer uses the Server Message Blocks (SMB) protocol; to send an email over the Internet, it uses the Simple Mail Transfer Protocol (SMTP).

After processing the request, the application layer protocol then passes the request down to the layer just below it. Each successive layer processes the request in some way, finally resulting in the conversion to appropriate signals at the bottom, or physical, layer. The network interface adapter in the computer then transmits the signals over the network medium.

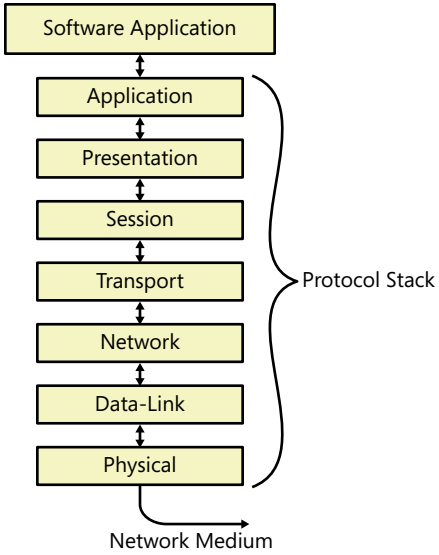

FIGURE 1-6 The seven layers of the OSI reference model.

When the signals arrive at their destination, they start at the bottom of the protocol stack and work their way up through the layers. The process is exactly the same, but reversed, as shown in Figure 1-7. Eventually, the message arrives at the corresponding application running on the destination computer.

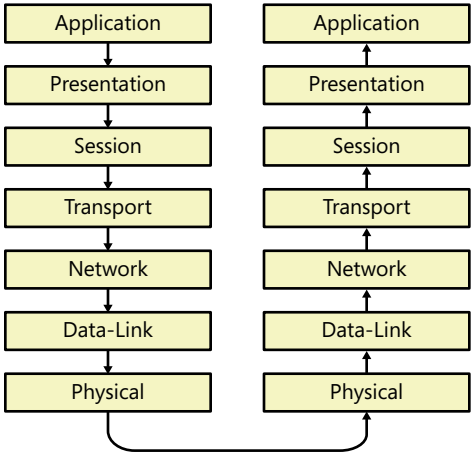

FIGURE 1-7 Protocols provide services to other protocols at adjacent layers, creating a path downward and upward through the stack.

The OSI reference model is defined in a standard document published in 1983 by the ISO called "The Basic Reference Model for Open Systems Interconnection." The same document was also published by the ITU-T as X.200. The standard divides the functions of the data networking process into the seven layers that form the protocol stack. The standard was

originally intended to be the model for an actual implementation of a new protocol stack, but this never materialized. Instead, the OSI model has come to be used with the existing network protocols as a teaching and reference tool.

It is important to understand that the actual protocols running in most network implementations do not conform exactly to the architecture of the OSI reference model. For example, the protocol stack in most computers does not consist of precisely seven protocols, with one operating at each of the seven defined layers. Some of the most commonly used protocols have functions that span multiple layers, whereas other layers might require two or more protocols to fulfill their functions.

Despite this lack of an exact correlation between the OSI model and actual networks, however, the IT industry often uses OSI terminology to describe networking functions.

EXAM TIP

Remembering the names of the OSI reference model layers, in the correct order, is an important part of your exam preparation. There are many mnemonics that students use to recall the layer names, ranging from the standard "All People Seem To Need Data Processing" to the silly "Programmers Do Not Throw Sausage Pizza Away" to the obscene, which you will have to discover for yourself.

Protocol Interaction

The protocols that make up the stack on a particular computer, despite being defined by different standards bodies and possibly created by different manufacturers, work together to provide all of the networking services required by the computer's applications and operating system. From a functional perspective, the stack usually is not redundant, meaning that when a protocol at one layer provides a particular service, the protocols at the other layers do not provide the exact same service, even if they are capable of doing so.

Protocols at adjacent layers of the stack provide services for the layer above and request services from the layer below, enabling data to make its way down (or up) through the layers. For example, when there is a choice of protocols at the transport layer, the protocol at the network layer below specifies which of the transport layer protocols the data it is passing upward should use. There is always, therefore, a definitive path that data should take upward or downward through the stack, depending on the services needed to transmit or receive the data.

Protocols operating at the same layer on different computers also provide complementary functions to each other. For example, if a protocol at a particular layer is responsible for encrypting data, the equivalent protocol at the same layer on the destination system must be responsible for decrypting it. In this way, the protocols at equivalent layers can be said to provide services to each other, as shown in Figure 1-8.

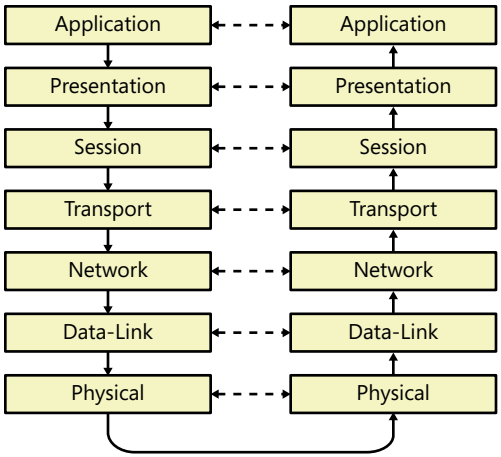

FIGURE 1-8 Protocols operating at the same layer on different computers provide complementary services to each other.

Data Encapsulation

The processing that occurs at each layer of the OSI reference model in most cases involves the application (or removal) of an additional block of data called a header to the *protocol data unit (PDU)* received from the adjacent layer. This process is called *data encapsulation*.

The process begins when an application creates a PDU containing a network access request at the application layer, as shown in Figure 1-9.

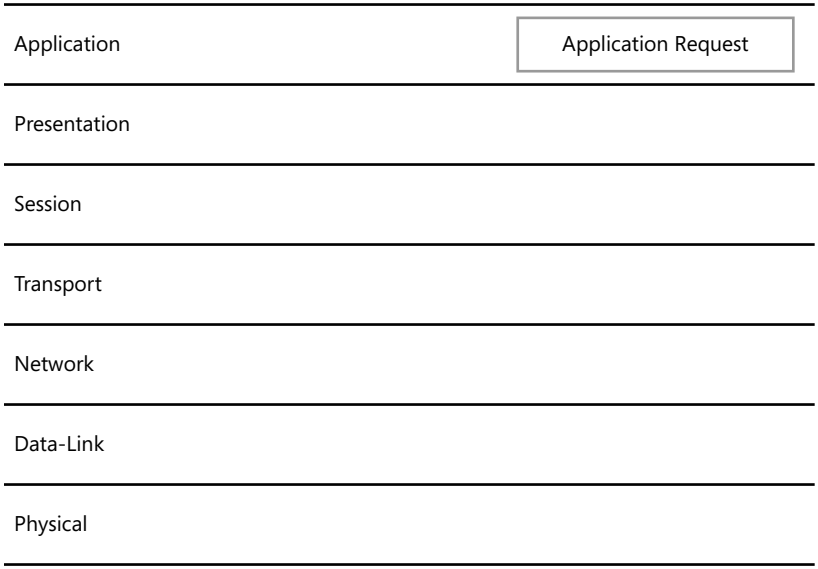

FIGURE 1-9 Data encapsulation: the application layer.

The application layer protocol then passes the PDU down to the transport layer. The transport layer protocol adds a header, creating a PDU of its own with the application layer data as the payload, as shown in Figure 1-10.

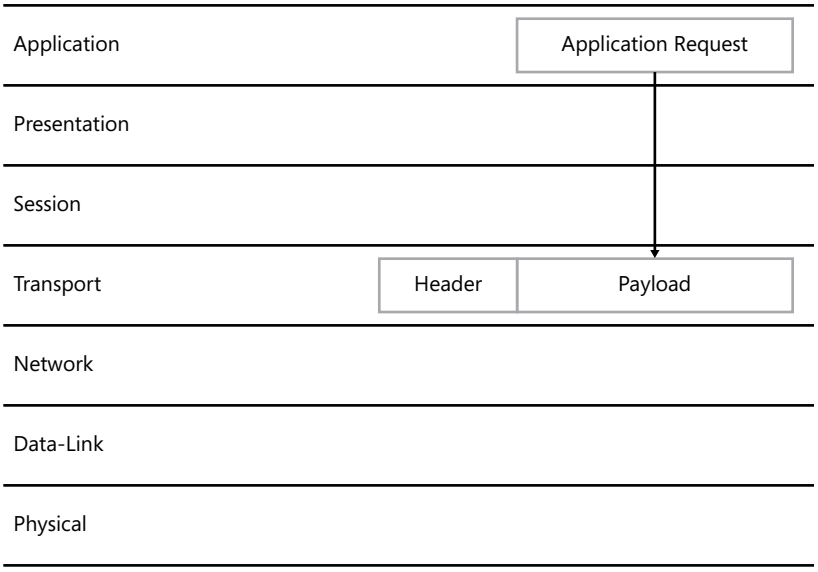

FIGURE 1-10 Data encapsulation: the transport layer.

NOTE **TCP/IP LAYERS**

The TCP protocol stack does not include separate protocols for the presentation and session layers. These functions are typically integrated into the application layer protocols. However, these two layers do provide a pass-through service that enables protocols at the layers above and below them to exchange data.

The transport layer protocol header consists of fields containing information that implements the protocol's various functions. The application layer PDU becomes the data field of the transport layer PDU.

When the transport layer protocol passes the PDU down to the network layer, the network layer protocol adds its own header in exactly the same way, as shown in Figure 1-11. Headers vary in size, depending on the number and nature of the functions implemented by the protocol.

The process is varied slightly at the data-link layer, which adds both a header and a footer to the network layer PDU, as shown in Figure 1-12.

The packet is now complete and ready for transmission over the network. All that remains is to convert the data into signals appropriate for transmission over the network medium.

The following sections examine, in general terms, the functions that occur at each layer of the OSI reference model. For more detailed studies of the various protocols running on today's computers, see the subsequent chapters in this book, as referenced at the end of each section.

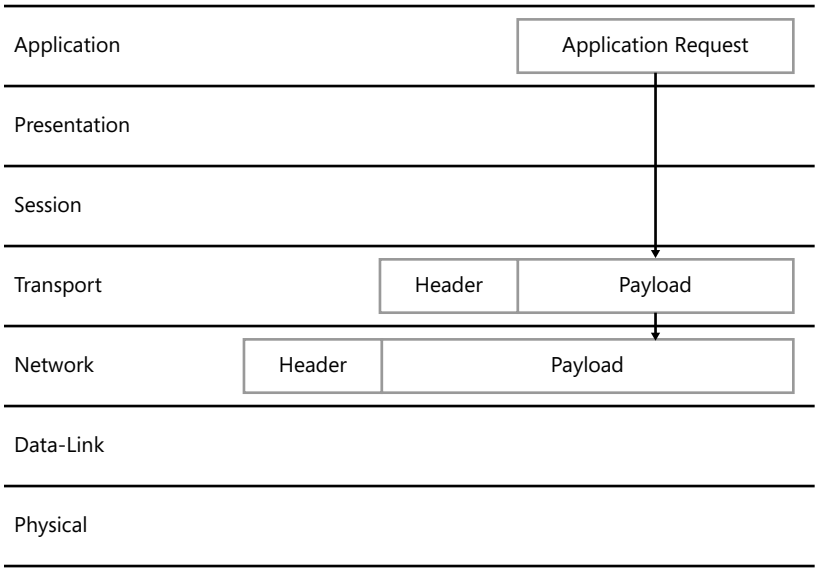

FIGURE 1-11 Data encapsulation: the network layer.

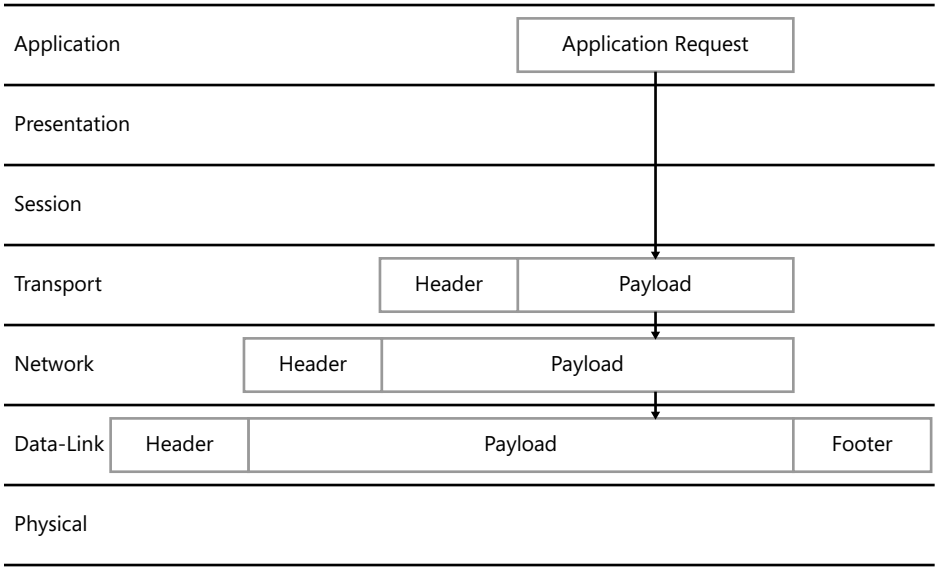

FIGURE 1-12 Data encapsulation: the data-link layer.

Quick Check

1. Name the layers of the OSI reference model at which protocols apply headers to outgoing packets.
2. On an Ethernet LAN, at which OSI model layer does a protocol apply a footer as well as a header?

Quick Check Answers

1. Transport, network, and data-link
2. Data-link

The Physical Layer

The physical layer, as shown in Figure 1-13, is at the bottom of the OSI reference model.

As the name implies, the physical layer is the layer that defines the hardware elements of the network, including the following:

- The network interface found in each computer or other device
- The nature of the signals used to transmit data over the network
- The characteristics of the network medium that carries the signals
- The physical topology of the network

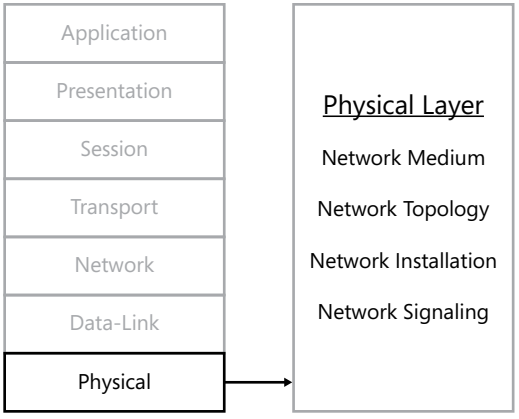

FIGURE 1-13 The physical layer of the OSI model.

Physical Layer Specifications for LANs

On a LAN, the physical layer specifications for the network are directly related to the selection of a data-link layer protocol. For example, if you elect to use Ethernet at the data-link layer, you must choose from among an assortment of physical layer specifications included in the Ethernet standards. These specifications dictate the types of cable you can use, the maximum lengths of the cables, and the number of devices you can connect to the LAN, among other things.

The data-link layer protocol standards do not necessarily contain all of the physical layer specifications needed to install a network, however. Some elements are defined in other standards.

One of the most commonly used physical layer specifications is the "Commercial Building Telecommunications Cabling Standard," published jointly by ANSI and the TIA/EIA as document 568-C. This document includes detailed specifications for installing cables for data networks in a commercial environment, including the required distances for cables from sources of electromagnetic interference and the pinouts for the cable connectors.

In most cases, organizations outsource large LAN cabling jobs to specialized cabling contractors, often the same ones responsible for wiring phone systems and other office infrastructure services. Any contractor you consider for a LAN cabling job should be very familiar with TIA/EIA 568-C and other such documents, including your local building codes.

Physical Layer Specifications for WANs

The physical and data-link specifications for LANs are closely associated because the LAN protocol is largely devoted to the sharing of the network medium among many computers. WAN links, however, are usually point-to-point connections between two—and only two—systems. As a result, WAN technologies typically use a relatively simple protocol at the data-link layer called the Point-to-Point Protocol (PPP), which does not contain any physical layer specifications. The WAN protocol can therefore have a completely independent hardware implementation at the physical layer.

Physical Layer Signaling

The final element found at the physical layer is the signaling method that systems use to transmit data over the network medium. The basic nature of the signals is, of course, determined by the network medium. Copper cables use electrical voltages, fiber optic cables use light pulses, and wireless networks can use several different radio and infrared signaling methods.

By the time data reaches the bottom of the protocol stack, it is a simple binary sequence—zeros and ones—and the signaling method is just a mechanism for encoding those binary digits. The actual signaling scheme that a network uses is not controllable by the network administrator; it is specified by the data-link layer protocol in the case of a LAN, or by the WAN technology. Therefore, although it might be interesting for a network administrator to know that Ethernet networks use a signaling scheme called Differential Manchester and how it works, it is not a subject that comes up in daily practice.

MORE INFO **LEARNING MORE ABOUT PHYSICAL LAYER PROTOCOLS**

For more detailed information about the physical layer specifications of specific types of networks, see the chapters listed in Table 1-1.

TABLE 1-1 Physical Layer Protocol Cross-References

Physical Layer Protocols	Chapter Coverage
EIA/TIA 568B	Chapter 2, "The Physical Layer"
IEEE 802.11a/b/g/n	Chapter 5, "Wireless Networking"
10Base-T / 100Base-TX / 1000Base-T, and others	Chapter 4, "The Data-Link Layer"
WAN Protocols	Chapter 10, "Wide Area Networking"

The Data-Link Layer

The data-link layer, as shown in Figure 1-14, is the second layer—or layer 2—of the OSI reference model.

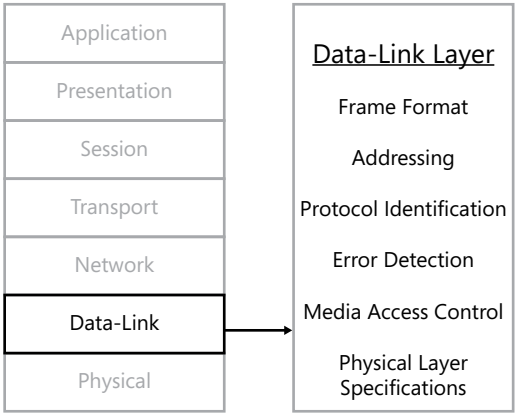

FIGURE 1-14 The data-link layer of the OSI model.

The protocol you elect to use at this layer is the primary factor that determines what networking hardware you purchase and how you install it. To implement a data-link layer protocol, you need the following hardware and software:

- **Network interface adapter** The hardware device that provides the computer with the actual connection to the network and implements some of the data-link layer protocol functions. Network adapters can be integrated into the computer's motherboard or take the form of internal expansion cards or external USB devices.
- **Network adapter driver** A software device driver that enables the computer to utilize the functions of the network interface adapter hardware.
- **Network cables (or other media) and other connecting hardware**
- **Network switches, hubs, or access points**

> ***NOTE*** **NETWORKING WITHOUT A HUB**
>
> **Although it is possible to connect computers together without a switch, hub, or access point—by using a crossover cable or an ad hoc wireless network—these are in most cases temporary solutions not suitable for a permanent installation.**

Most of these components are designed specifically for a certain data-link layer protocol, and more specifically, for a protocol running at a certain speed. For example, you might decide to use Ethernet at the data-link layer, but when purchasing hardware, you must be careful to distinguish between products supporting regular Ethernet, Fast Ethernet, and Gigabit Ethernet. Most of the newer products on the market are backward compatible with older, slower devices, but each branch of your network will only be as fast as its slowest link.

It is the network interface adapter in each computer, in combination with the network adapter driver, that actually implements the data-link layer protocol. Some of the data-link layer protocol functions are performed by the adapter independently, before incoming data is passed to the computer and before outgoing data leaves it. Other functions are performed by the driver after the adapter passes incoming data to the computer and before the computer passes outgoing data to the adapter. Generally speaking, higher-end—and higher-priced—adapters contain processors that perform more of the networking functions on board, rather than leaving them to the system processor.

Data-Link Layer Standards for LANs

In the case of LANs, the data-link layer protocols in most common use today are Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11). The standards for these protocols consist of the following elements:

- A frame format
- A media access control mechanism
- Physical layer specifications

These components are discussed in the following sections.

FRAME FORMAT

Data-link layer LAN protocols use the term *frame* to refer to the protocol data unit they create by using the information they receive from the network layer. This is largely because the data-link layer protocol adds both a header and a footer to the network layer PDU.

The data-link layer frame typically performs the following functions:

- **Addressing** The header and footer that the data-link layer protocol applies functions as the outermost envelope in the figurative mailing of a packet. The header contains addresses identifying the packet's sending and receiving systems. These addresses—known as "hardware addresses" or "media access control (MAC) addresses"—are 6-byte hexadecimal strings hard-coded into the computers' network interface adapters.

IMPORTANT **PROTOCOL DATA UNITS**

The protocols operating at the different layers of the OSI reference model use different terms to refer to the PDUs they create. The terms most often found in the networking literature are listed in Table 1-2.

TABLE 1-2 PDU Terminology

OSI Model Layer	Protocol	PDU Terminology
Data-Link	Ethernet	Frame
Network	Internet Protocol	Datagram
Transport	User Datagram Protocol	Datagram
Transport	Transmission Control Protocol	Segment
Application	Various	Message

The term "packet" is generic and can refer to the PDU at any stage of the data encapsulation process.

NOTE **DATA-LINK LAYER COMMUNICATIONS**

It is important to understand that data-link layer protocols are limited to communication with systems on the same subnet. A computer might transmit a packet to a destination on another LAN, but the data-link layer protocol is only involved in local subnet communications. The hardware address in a data-link layer protocol header always refers to a system on the same subnet. This typically means that the data-link layer protocol carries the packet only as far as the nearest router. Inside the router, a network layer protocol assumes responsibility for delivering the packet to its final dsestination, as discussed later in this chapter.

- **Network layer protocol identification** The data-link layer protocol header contains a code that specifies which network layer protocol is encapsulated within the frame. This way, when the packet works its way up the protocol stack on the receiving system, the data-link layer protocol can determine where to pass the encapsulated data.
- **Error detection** The transmitting system performs a *cyclical redundancy check (CRC)* calculation on the data in the frame and appends the result to the packet as a footer. When the packet arrives at the destination, the receiving system performs the same calculation. If the results do not match, the receiving system assumes that a transmission error has occurred and discards the packet.

MEDIA ACCESS CONTROL

When computers are connected to a shared baseband medium, as on a typical LAN, it is possible for two systems to transmit packets at exactly the same time. This results in a *collision*, causing the corruption and loss of both packets. One of the primary functions of a data-link layer protocol on a LAN is to prevent, minimize, or handle collisions. To do this, the protocol includes a *media access control (MAC)* mechanism. The MAC mechanism provides each of the computers on the LAN with regular opportunities to transmit its data.

On modern switched Ethernet networks, the switches provide each pair of devices with a dedicated channel, eliminating the need for computers to share the network medium, and consequently reducing the need for a MAC mechanism. Media access control is also not so elaborate a function on WANs because there are only two systems involved, and they can simply take turns transmitting.

PHYSICAL LAYER SPECIFICATIONS

One of the main reasons why LAN data-link layer protocol standards include physical layer specifications is to support the protocol's MAC mechanism. If, for example, the cables are too long on an Ethernet network, the systems will be unable to detect packet collisions when they occur, and collision detection is one of the critical elements of the Ethernet MAC mechanism. Excessively long cables can also result in signal degradation and the increased likelihood of signal interference.

MORE INFO **LEARNING MORE ABOUT DATA-LINK LAYER PROTOCOLS**

For more detailed information about the data-link layer specifications of specific types of networks, see the chapters listed in Table 1-3.

TABLE 1-3 Data-Link Layer Protocol Cross-References

Data-Link Layer Protocols	Chapter Coverage
Ethernet	Chapter 4, "The Data-Link Layer"
IEEE 802.11	Chapter 5, "Wireless Networking"
Point-to-Point Protocol (PPP)	Chapter 4, "The Data-Link Layer"

The Network Layer

The network layer, as shown in Figure 1-15, is the third layer—or layer 3—of the OSI reference model.

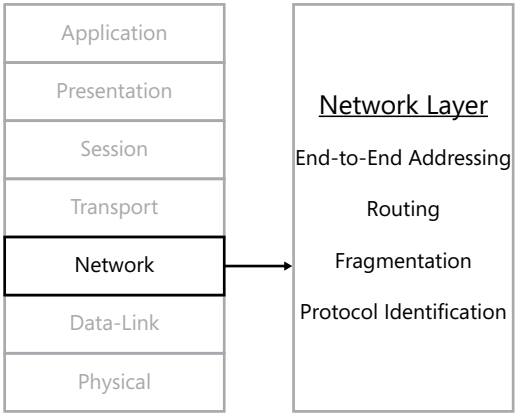

FIGURE 1-15 The network layer of the OSI model.

Just as the data-link layer protocol is responsible for communications within a LAN or WAN, the network layer protocol is primarily responsible for end-to-end communications between a packet's source and its ultimate destination.

For example, when a workstation on a corporate network accesses a webpage on the Internet, the stack's data-link protocol is only responsible for getting the packets to the router on the local network. The network layer protocol, however, is responsible for getting the packets to the web server on the Internet, which might involve transmission through dozens of different networks. Those intermediate networks might use different data-link layer protocols, but they all use the same network layer protocols.

At one time, there were several different end-to-end protocols that systems could use at the network layer. Today, however, the TCP/IP protocol suite is nearly ubiquitous, due to its role as the backbone of the Internet, and the end-to-end network layer protocol in that suite is the *Internet Protocol (IP)*.

EXAM TIP

Obsolete protocols at the network layer include Internetwork Packet Exchange (IPX), an end-to-end protocol developed for use with Novell NetWare networks, and NetBEUI, a non-routable file and printer sharing protocol used in early versions of the Windows operating system. These protocols were formerly part of the Network+ curriculum, but the latest version of the exam covers only IP.

As noted earlier in this chapter, the network layer protocol encapsulates data it receives from the layer above by applying a header. The functions performed by the IP header are described in the following sections.

Addressing

A network layer protocol header contains a source address and a destination address—just as a data-link layer protocol header does. However, the difference between the two is that the network layer destination address identifies the final recipient of the packet, whether it is on the local network or another network thousands of miles away.

Although some of the obsolete network layer protocols used the same hardware addresses as Ethernet and other data-link layer protocols, IP has its own independent addressing system that uses 32-bit or 128-bit numerical strings. IP addresses identify both the system itself and the network on which the system is located.

> ***NOTE*** **IP VERSIONS**
>
> **The IP protocol is currently in the midst of a lengthy transition from version 4 (IPv4) to version 6 (IPv6). IPv4 uses 32-bit addresses, but in IPv6, the address space is expanded to 128 bits. For more details, see Chapter 6, "The Network Layer."**

Fragmentation

When a network layer protocol encapsulates the data it receives from the layer above, it creates a datagram that will remain intact as a PDU until it reaches its final destination. However, that datagram might have to pass through dozens of different data-link layer networks on its journey. These networks might have different properties, including various maximum frame sizes. When a computer has to transmit a datagram that is too large for the data-link layer network, it splits the datagram into fragments and transmits each one in a separate data-link layer frame.

Depending on the nature of the intervening networks, individual fragments might be fragmented again before they complete their journey. It is not until the fragments reach their final destination—that is, the system identified by the destination address in the network layer protocol header—that the network layer protocol reassembles them into the original datagram.

Routing

As noted earlier in this chapter, routers are the devices that connect LANs and WANs together, forming an internetwork. A router has a minimum of two network interfaces, enabling it to communicate on two networks simultaneously. Routing is the process of directing a network layer datagram from its source to its final destination, while attempting to use the most efficient possible path through the internetwork.

There are two types of systems in internetwork communications: end systems and intermediate systems. *End systems* are the sources or destinations of individual packets, that is, the systems whose addresses appear in the network layer protocol header. *Intermediate systems* are the routers that pass datagrams from one network to another.

When an end system processes a packet, all seven layers of the OSI reference model are involved. When a packet passes through an intermediate system, it enters through one network interface, travels up through the stack only as high as the network layer, and then travels down again to the other network interface and out over the second network, as shown in Figure 1-16. In this way, the routers pass the datagram from network to network until it reaches the destination network—that is, the network on which the destination system is located.

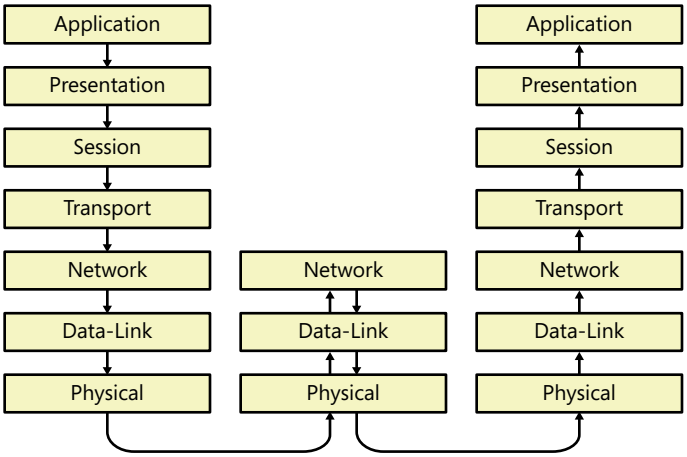

FIGURE 1-16 The network layer protocol in a router accepts incoming packets and transmits them to the next stop on their journey.

To direct packets to their destinations, routers maintain information about the networks around them in a routing table, which they store in memory. It is possible for administrators to manually create entries in a routing table, but it is more common for the systems to compile the routing information themselves by using specialized routing protocols. An intermediate system only has direct knowledge of the networks to which it is directly connected, but by using a routing protocol, it can share that information with other routers and receive information about distant networks in return.

By sharing this information, each router creates entries in its routing table. Each entry contains the address of another network, the address of an intermediate system providing access to that network, and a value called a metric that rates the comparative efficiency of that particular route. When a router receives an incoming datagram, it reads the destination address from the network layer protocol header, and checks that address against its routing table. If the router has information about the destination network, it forwards the datagram to the next router in its path.

Each journey from one router to another is referred to as a *hop*. In many cases, the efficiency of a route is measured by the number of hops—that is, intervening routers—between the source and the destination. On a corporate internetwork, the routing process is often very simple, but on the Internet, the path of a datagram from source to destination can consist of dozens of hops.

Network Layer Error Detection

Data-link layer protocols for LANs typically have an error detection mechanism, as described earlier in this chapter. However, this mechanism only provides protection from transmission errors on the LAN; it does not provide end-to-end protection for the datagram's entire journey from source to destination.

Network layer protocols might therefore have their own error detection mechanisms, but this can also be the province of the transport layer. In the case of IP, the network layer protocol does have end-to-end error protection, but only for the contents of the IP header, not for the payload carried inside the datagram.

Transport Layer Protocol Identification

The network layer protocol header contains a code that identifies the transport layer protocol encapsulated in the packet, just as the data-link layer header identifies the network layer protocol. This ensures that the datagram arriving at the network layer is passed to the appropriate protocol at the transport layer.

When IP is the network layer protocol, the protocol identification code in the header typically references either the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP).

MORE INFO **LEARNING MORE ABOUT NETWORK LAYER PROTOCOLS**

For more detailed information about the network layer specifications of specific types of networks, see the chapters listed in Table 1-4.

TABLE 1-4 Network Layer Protocol Cross-References

Network Layer Protocols	Chapter Coverage
Internet Protocol	Chapter 6, "The Network Layer" Chapter 7, "Routing and Switching"

The Transport Layer

The protocols at the transport layer of the OSI reference model, as shown in Figure 1-17, work in conjunction with the network layer protocol to provide a unified quality of service required by the application that is making use of the network.

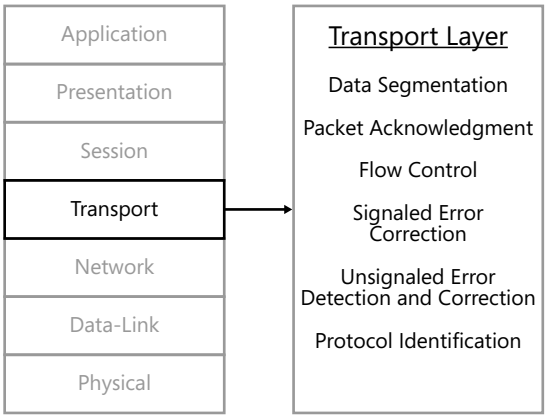

FIGURE 1-17 The transport layer of the OSI model.

Because they work together providing the services that the application needs, the network and transport layer protocols are nearly always created by the same team of standards-makers and are specifically designed to complement each other.

The protocol combinations at the network and transport layers are so intimately linked that the protocol suites they come from are frequently named for them. The TCP/IP suite takes its name from the combination of the *Transmission Control Protocol (TCP)*, one of its transport layer protocols, and IP, at the network layer. In the same way, the once-dominant protocol suite for the Novell NetWare operating system was called IPX/SPX, for the Internetwork Packet Exchange and Sequenced Packet Exchange protocols, at the network and transport layers, respectively.

The TCP/IP suite also provides a second protocol at the transport layer, called the *User Datagram Protocol (UDP)*. Generally speaking, the TCP protocol provides a wide range of services, at the cost of a great deal of data transmission overhead. UDP provides a minimal service, with much less overhead.

Connection-Oriented and Connectionless Protocols

There are two basic types of protocols at the transport layer—and at the network layer, for that matter—connection-oriented and connectionless. A *connection-oriented protocol* is one in which the two communicating systems establish a connection between themselves before they begin transmitting any data. The connection establishment process ensures that the two systems are operational and ready to communicate.

TCP is a connection-oriented protocol, and the procedure it uses to establish a connection with the destination system is called a three-way handshake. When the systems have finished exchanging data, they perform another handshake to terminate the connection.

TCP also provides other services to the applications that are running in the upper layers, including data segmentation, packet acknowledgment, flow control, and end-to-end error detection and correction. To implement all of these services, TCP needs a large header, and

the handshakes force the systems to transmit additional packets. This is all in addition to the actual application layer data they have to transmit.

The transmission overhead for a connection-oriented protocol is therefore quite high, making it suitable only for applications that require its extensive services. In most cases, applications that use TCP are those that require bit-perfect data transmission, such as the transfer of a program file. If even one bit of a program is incorrect, it won't run properly.

Some applications do not require bit-perfect transmission, however. For example, when a system is streaming video over a network, the loss of a few bits might cause a momentary degradation of the picture quality, but it will not cause the application to fail. This type of data exchange can use a *connectionless protocol* instead, one that does not require a connection establishment process and one that provides a minimum of additional services. Because connectionless protocols do not provide additional services, their headers are smaller, and their transmission overhead is much lower than that of a connection-oriented protocol. For example, though the TCP header is typically 20 bytes, the header of the connectionless UDP protocol is only 8 bytes.

NOTE **CONNECTIONLESS PROTOCOLS**

The network layer IP protocol is also considered to be connectionless. Because nearly all data packets use IP at the network layer, it is sensible to use a connectionless protocol at that layer, and reserve the optional connection-oriented services for the transport layer.

A typical UDP transaction consists of only two messages: a request and a reply. In this type of transaction, the reply functions as a tacit acknowledgment, so there is no need for a connection establishment process or an elaborate packet acknowledgment mechanism.

Transport layer protocols can provide a variety of services to applications. In the TCP/IP suite, UDP provides minimal service, and TCP is considered to be the full-service protocol. In addition to connection orientation, TCP also provides the services described in the following sections.

Packet Segmentation

Applications generate data without considering the nature of the network at all, so one of the primary functions of the transport layer is to split the application layer data into segments of a size suitable for transmission. The protocol assigns numbers to the segments, which the receiving system uses to identify specific packets for acknowledgment, retransmission, and reassembly.

NOTE **SEGMENTATION**

Transport layer segmentation is a completely separate process from the fragmentation that occurs at the network layer. Data might end up being both segmented and fragmented during the course of its transmission to a specific destination.

Packet Acknowledgment

TCP is often referred to as a "reliable" protocol because it provides guaranteed delivery, a service that takes the form of acknowledgment messages transmitted by the receiving system back to the sender. Although there have at times been protocols that generated a separate acknowledgment message for each individual transmitted packet, TCP is able to acknowledge multiple segments with one acknowledgment, which helps to reduce the protocol's overhead.

> ***NOTE*** **RELIABLE PROTOCOLS**
>
> **In this case, "reliable" is a technical term referring to the fact that each packet transmitted by using the TCP protocol has been acknowledged by the recipient, and has been verified as having been transmitted without error. It is not an indication that other protocols—such as UDP—cannot be trusted to deliver their data.**

Flow Control

Flow control is a mechanism that enables a receiving system to regulate the rate at which the sending system transmits data. If the sender transmits too many packets in a specified period of time, the buffer on the receiving system might fill up, preventing it from receiving any more packets until the buffer empties.

When this occurs, the only alternative is for the receiving system to discard some of the packets. The transmitting system will eventually have to resend the missing packets, but the error detection and correction processes reduce the efficiency of the connection.

To prevent this condition from continuing, the receiving system sends a series of flow control messages to the sender, ordering it to reduce its transmission rate. When the buffer empties, the receiver can order the sender to speed up again.

Error Detection and Correction

In TCP/IP, the transport layer protocol is the only protocol that provides complete end-to-end error detection and correction for the entire packet, including the data passed down from the application layer. The data-link layer protocol can detect errors, but it cannot correct them by retransmitting packets. Instead, the data-link layer protocol passes the error information up the stack—such messages are called signaled errors—and the transport layer protocol takes responsibility for the error correction process.

The transport layer protocol also performs its own CRC check on the entire packet. Errors that the protocol discovers itself are called unsignaled errors. The protocol corrects the errors by manipulating the packet acknowledgment messages it transmits back to the sender. When the sender does not receive an acknowledgment for each packet within a certain period of time, the retransmission process is automatic.

Application Layer Protocol Identification

To maintain the integrity of the protocol stack, the transport layer protocol must include codes in its header identifying the applications responsible for the data on the source and destination computers. These codes are called port numbers, and they identify specific protocols running at the application layer.

MORE INFO **LEARNING MORE ABOUT TRANSPORT LAYER PROTOCOLS**

For more detailed information about transport layer protocols, see the chapters listed in Table 1-5.

TABLE 1-5 Transport Layer Protocol Cross-References

Transport Layer Protocols	Chapter Coverage
Transmission Control Protocol (TCP)	Chapter 8, "The Transport Layer"
User Datagram Protocol (UDP)	Chapter 8, "The Transport Layer"

The Session Layer

The boundary between the transport layer and the session layer, as shown in Figure 1-18, is a major division in the protocol stack.

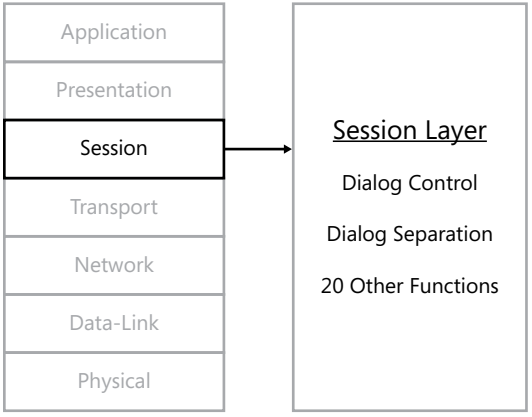

FIGURE 1-18 The session layer of the OSI model.

It is common for administrators to refer to the session, presentation, and application layers collectively as "the upper layers," because this is where the real-world protocol implementations start to bear less of a resemblance to the OSI model.

For example, it is common for an application layer protocol to provide services associated with the session and/or presentation layers, as well as the application layer. Although session layer protocols do exist, they are typically not independent entities like the protocols you find at the lower layers. Instead, they are integrated into larger networking components.

REAL WORLD **NETBIOS AND THE SESSION LAYER**

NetBIOS (network basic input/output system) provides session layer services to Windows workgroup (that is, non-Active Directory) networks. For example, the 16-character computer names assigned to all workgroup computers are actually NetBIOS names. Even computers that are connected to an Active Directory Domain Services (AD DS) network have NetBIOS equivalents to their AD DS names.

However, in the current versions of Windows, NetBIOS takes the form of an application programming interface (API), not a networking protocol. Workgroup networks use a hybrid protocol called NetBIOS Over TCP/IP (NetBT) to transmit data by using the standard TCP/IP network and transport layer protocols. Therefore, you can think of components such as NetBIOS as session layer services, but they are distinctly different from the lower layer protocols in their implementations.

The session layer is also the dividing line where computers leave behind all concerns for efficient transmission of data across the network. Protocols at the session layer and above do not provide any of the services—such as addressing, routing, and error correction—needed to get data from point A to point B. These functions are strictly relegated to the lower-layer protocols.

Because of its name, many people associate the session layer exclusively with security functions, such as the network logon process that establishes a "session" between two computers. In fact, the session layer does not have a single primary function, unlike the lower layers.

The session layer is more of a "toolbox" containing a variety of functions. The OSI model standard defines 22 services for the session layer, many of which are concerned with the ways in which networked systems exchange information. Many of these services are quite obscure to everyone except application developers.

Some of the most important session layer functions are concerned with the exchange of data by the two end systems involved in a connection. However, the session layer is not concerned with the nature of the data being exchanged, but rather with the exchange process itself, which is called a *dialog*. Maintaining an efficient dialog between connected computers is more difficult than it might appear, because requests and replies can cross each other in transit, leaving the computers in an unknown state. The session layer functions include mechanisms that help the systems maintain an efficient dialog. The most important of these services are dialog control and dialog separation.

Dialog Control

The exchange of information between two systems on the network is a dialog, and dialog control is the selection of a mode that the systems will use to exchange messages. When the dialog begins, the systems can choose one of two modes: two-way alternate (TWA) mode or two-way simultaneous (TWS) mode. In TWA mode, the two systems exchange a data token, and only the computer in possession of the token is permitted to transmit data. This eliminates problems caused by messages that cross in transit. TWS mode is more complex, because there is no token and both systems can transmit at any time, even simultaneously.

Dialog Separation

Dialog separation is the process of creating checkpoints in a data stream that enable communicating systems to synchronize their functions. The difficulty of checkpointing depends on whether the dialog is using TWA or TWS mode. Systems involved in a TWA dialog perform minor synchronizations that require only a single exchange of checkpointing messages, but systems using a TWS dialog perform a major synchronization using a major/activity token.

MORE INFO **LEARNING MORE ABOUT SESSION LAYER PROTOCOLS**

For more detailed information about session layer protocols, see the chapters listed in Table 1-6.

TABLE 1-6 Session Layer Protocol Cross-References

Session Layer Protocols	Chapter Coverage
Layer 2 Tunneling Protocol (L2TP)	Chapter 10, "Wide Area Networking"
Point-to-Point Tunneling Protocol (PPTP)	Chapter 10, "Wide Area Networking"

The Presentation Layer

The presentation layer, as shown in Figure 1-19, is the simplest of the seven in the OSI model.

For the most part, the presentation layer functions as a simple pass-through connecting the application layer to the session layer. For each of the 22 session layer functions defined in the OSI model standard, there is a corresponding pass-through function defined at the presentation layer. This is so that an application layer protocol can access any of the session layer services by sending a request to the presentation layer, which passes it down to the correct session layer function.

In addition to the pass-through services, the presentation layer also provides a syntax translation service that enables two computers to communicate, despite their use of different bit-encoding methods. This translation service also enables systems using compressed or encrypted data to communicate with each other.

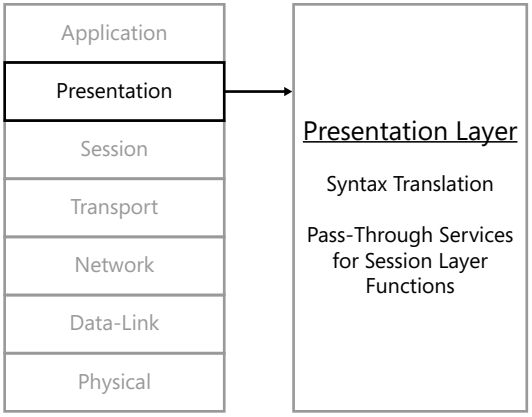

FIGURE 1-19 The presentation layer of the OSI model.

Here again, as in the session layer, the presentation layer standards do not take the form of networking protocols. For example, two of the most prominent bit-encoding methods, ASCII (American Standard Code for Information Interchange) and EBCDIC (Extended Binary Coded Decimal Interchange Code), are simply tables of binary codes equivalent to the standard US English character set.

The translation occurs in two stages. The presentation layer on the sending system translates the message from its native form, which is called an *abstract syntax*, to a *transfer syntax*, which is a common syntax agreed upon by the two connected end systems. After it receives the message, the destination system translates the message from the transfer syntax to that computer's own abstract syntax.

The Application Layer

The application layer, at the top of the protocol stack, as shown in Figure 1-20, is the entrance point that programs running on a computer use to access the network protocol stack.

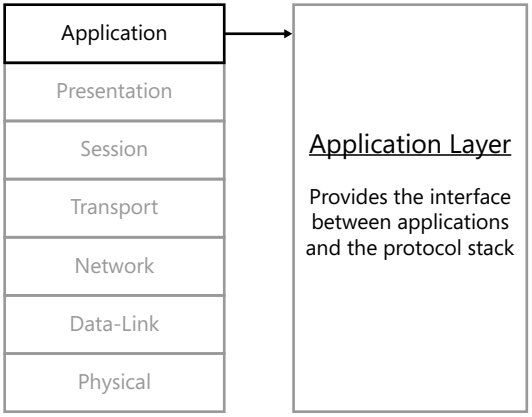

FIGURE 1-20 The application layer of the OSI model.

MORE INFO **LEARNING MORE ABOUT APPLICATION LAYER PROTOCOLS**

There are many application layer protocols—more than at any other layer of the OSI model. For more detailed information about these protocols, see the chapters listed in Table 1-7.

TABLE 1-7 Application Layer Protocol Cross-References

Application Layer Protocols	Chapter Coverage
Dynamic Host Configuration Protocol (DHCP)	Chapter 6, "The Network Layer"
Domain Name System (DNS)	Chapter 6, "The Network Layer"
File Transfer Protocol (FTP)	Chapter 9, "The Application Layer"
Hypertext Transfer Protocol (HTTP)	Chapter 9, "The Application Layer"
Internet Message Access Protocol (IMAP)	Chapter 9, "The Application Layer"
Network File System (NFS)	Chapter 9, "The Application Layer"
Network Time Protocol (NTP)	Chapter 9, "The Application Layer"
Open Shortest Path First (OSPF)	Chapter 7, "Routing and Switching"
Post Office Protocol version 3 (POP3)	Chapter 9, "The Application Layer"
Real-time Transport Protocol (RTP)	Chapter 9, "The Application Layer"
Routing Information Protocol (RIP)	Chapter 7, "Routing and Switching"
Secure Shell (SSH)	Chapter 11, "Understanding Network Security"
Session Initiation Protocol (SIP)	Chapter 9, "The Application Layer"
Simple Network Management Protocol (SNMP)	Chapter 12, "Network Management"
Simple Mail Transfer Protocol (SMTP)	Chapter 9, "The Application Layer"
Telnet	Chapter 9, "The Application Layer"
Trivial File Transfer Protocol (TFTP)	Chapter 9, "The Application Layer"

In nearly all cases, the application layer protocol is not the actual application that the user sees; it is rather an application programming interface (API) call or protocol that provides a service to the application. All of the processes operating at the other OSI model layers are triggered when a program calls for the services of an application layer protocol. For example, an email client application provides users with tools to create a message, but it does not have actual networking capabilities built into it. When the client is ready to send the email message, it calls a function of the Simple Mail Transfer Protocol (SMTP), which is the application

layer protocol that most email programs use. SMTP then generates an appropriately formatted message and starts it on its way down through the layers of the protocol stack.

Application layer protocols often include session and presentation layer functions, which is why there are few dedicated presentation or session layer protocols. As a result, a typical packet is encapsulated four times before being transmitted over the network, by protocols running at the application, transport, network, and data-link layers.

Applications and application layer protocols are integrated to varying degrees. In the case of the email client mentioned earlier, the client program is a separate application, and SMTP is implemented as part of the TCP/IP protocol suite. However, in other cases, the application layer protocol is indistinguishable from the application. For example, FTP and Telnet implementations contain both user interface and application layer interface components.

The TCP/IP Model

The development of the TCP/IP protocols began years before the documents defining the OSI reference model were published, but the protocols conform to a layered model in much the same way. Instead of the seven layers used by the OSI model, the TCP/IP model—sometimes called the Department of Defense (DoD) model—has four layers, which are defined in RFC 1122, "Requirements for Internet Hosts – Communication Layers." The TCP/IP model layers, in comparison with those of the OSI model, are shown in Figure 1-21.

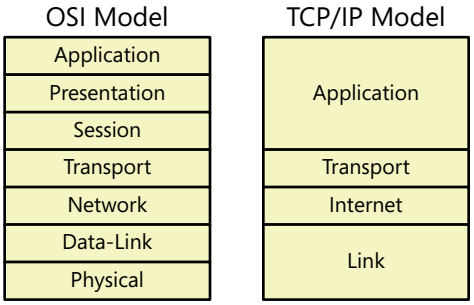

FIGURE 1-21 The four TCP/IP model layers, compared with the seven-layer OSI reference model.

EXAM TIP

The N10-005 revision of the Network+ exam objectives released in 2011 adds the TCP/IP model and specifically requires students to compare its layers with those of the OSI model. Be careful to distinguish between the two models, and familiarize yourself with the differences between the corresponding layers.

The TCP/IP model layers—even those with the same names—are not exactly analogous to the OSI model layers, nor are the purposes of the models the same. The OSI model is intended to be a guide for the creation of networking protocols, whereas the TCP/IP model is a representation of protocols that already exist.

The four TCP/IP layers, from bottom to top, are discussed in the following sections.

NOTE **THE TCP/IP MODEL**

Although the functionality defined in the four layers of the TCP/IP protocol stack can encompass the OSI model from data-link to application layer, the TCP/IP protocol stack is hardware-independent by design and therefore does not include physical layer specifications.

The Link Layer

The link layer, like the data-link layer of the OSI model, defines the mechanism for moving packets between two devices on the same local subnet, referred to as a *link* in TCP/IP terminology. The TCP/IP protocol suite includes two link layer protocols: Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP). SLIP is now all but obsolete in PC networking, and PPP is used for direct connections between nodes, as in most WAN technologies.

Despite being roughly analogous to the OSI data-link layer, the TCP/IP link layer does not include physical specifications of any kind, nor does it include complex LAN protocols such as Ethernet. Therefore, on many TCP/IP networks, the protocol operating at the link layer might not be part of the TCP/IP suite.

NOTE **IETF STANDARDS**

The IETF develops the TCP/IP specifications using a philosophy different from that of the other organizations responsible for networking standards, such as the ISO and the IEEE. Unlike the OSI reference model document, for example, the specification describing the TCP/IP model is informal, and deliberately omits certain aspects of the protocol stack, enabling implementers to exercise greater freedom in their designs. The omission of any specific functionality of the link is an excellent example of this philosophy.

When a TCP/IP system uses PPP at the link layer, the protocol stack assumes the presence of a network medium providing the physical connection, because PPP also does not include physical layer specifications. When the link layer functionality is provided by a non-TCP/IP protocol, as on a LAN, TCP/IP assumes the presence of both a valid network medium and a protocol that provides an interface to that medium.

Although the TCP/IP standards do not define the link layer protocol itself on a LAN, there are TCP/IP standards that define the interaction between the internet layer protocol (IP) and the protocol that provides the link layer functionality. For example, to reconcile the MAC addresses of network interface adapters with the IP addresses used at the internet layer, systems use a protocol in the TCP/IP suite called the Address Resolution Protocol (ARP). In addition, the use of Ethernet with TCP/IP is governed by the following standards:

- **RFC 826** "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48-bit Ethernet Address [sic] for Transmission on Ethernet Hardware"
- **RFC 894** "A Standard for the Transmission of IP Datagrams over Ethernet Networks"

The Internet Layer

The TCP/IP internet layer is exactly equivalent to the network layer of the OSI reference model. As in the OSI model, the Internet Protocol (IP) is the primary protocol operating at this layer. IPv4 and IPv6 provide connectionless services to the protocols operating at the transport layer above, including data encapsulation, routing, and addressing. Two additional protocols, the Internet Control Message Protocol (ICMP) and the Internet Group Management Protocol (IGMP), also operate at the internet layer, as do some specialized dynamic routing protocols.

EXAM TIP

In this context, the term "internet" is a generic reference to an internetwork and uses a lowercase "i," as opposed to the public, packet-switching Internet, with an uppercase "I." Be careful not to confuse the two.

The Transport Layer

The TCP/IP transport layer is roughly equivalent to the transport layer in the OSI model, in that it contains the same two protocols: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP and UDP provide connection-oriented and connectionless data transfer services, respectively, to application layer protocols.

The TCP/IP transport layer can in some ways be said to encompass some of the functionality attributed to the OSI session layer as well as the transport layer in the OSI model, but not in every case. Windows systems, for example, use TCP/IP to carry the session layer NetBIOS messages they use for their file and printer sharing activities.

This is one illustration of how the layers of the TCP/IP model are roughly equivalent to those of the OSI model, but not precisely so. Administrators now use these models more as pedagogical and diagnostic tools than as guidelines for protocol development and deployment; they sometimes do not hold up to strict comparisons of the various layers' functions with the actual working protocols.

The Application Layer

The TCP/IP application layer is analogous to the application, presentation, and session layers of the OSI model. However, the TCP/IP standards do not require application layer protocols to implement the functions of all three layers. In some cases, two or three separate protocols can provide these functions, whereas other application layer protocols are monolithic in their design.

The TCP/IP protocols at the application layer take two distinct forms, as follows:

- **User protocols** Provide services directly to users, as in the case of the File Transfer Protocol (FTP) and Telnet protocols
- **Support protocols** Provide common system functions, as in the case of the Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) protocols

As in the OSI model, application layer protocols are not concerned with transport; they assume the existence of a functional transport mechanism at the layers below, without duplicating any of the lower-layer services.

Quick Check

1. **Which layer of the OSI reference model does not have a corresponding layer in the TCP/IP model?**
2. **What two TCP/IP protocols operate at the link layer of the TCP/IP model?**

Quick Check Answer

1. **The physical layer**
2. **SLIP and PPP**

Exercise

The answers for this exercise are located in the "Answers" section at the end of this chapter.

Match the following terms to the OSI model layers with which they are most commonly associated.

1. frame
2. port
3. fragment
4. end-to-end addressing
5. segment
6. footer

7. message
8. media access control
9. flow control
10. LAN addressing
11. dialog separation
12. physical topology
13. abstract syntax
14. routing
15. packet acknowledgment

Chapter Summary

- A local area network (LAN) is a group of computers or other devices that share a common medium, such as a particular type of cable or wireless technology. A wide area network (WAN) is a group of computers connected by a longer distance communication technology provided by a third-party service provider, such as a telephone company.
- Protocols are languages that operate at various levels of the networking software on each computer. Two computers on the same network must use the same protocols to communicate.
- The combination of protocols running at the same time in a network implementation is called the network protocol stack. The Open Systems Interconnection (OSI) reference model is a theoretical example of a networking stack, which networking students and administrators use to categorize and define a computer's various networking functions.
- Protocols at adjacent layers of the networking stack provide services for the layer above and request services from the layer below, enabling data to make its way down (or up) through the layers.
- The processing that occurs at each layer of the OSI reference model in most cases involves the addition (or corresponding removal) of an extra block of data called a header. This process is called data encapsulation.
- The physical layer is the layer that defines the hardware elements of the network.
- The protocol you elect to use at the data-link layer is the primary factor that determines what networking hardware you will need to purchase and how you install it.
- The network layer protocol is primarily responsible for end-to-end communications between a packet's source and its ultimate destination.

- The protocols at the transport layer of the OSI model work in conjunction with the network layer protocol to provide a unified quality of service required by the application making use of the network.
- The application layer, at the top of the protocol stack, is the entrance point that programs running on a computer use to access the network protocol stack.

Chapter Review

Test your knowledge of the information in Chapter 1 by answering these questions. The answers to these questions, and the explanations of why each answer choice is correct or incorrect, are located in the "Answers" section at the end of this chapter.

1. Which of the following OSI model layers provides end-to-end error detection and correction for the entire packet?
   - **A.** Data-link
   - **B.** Network
   - **C.** Transport
   - **D.** Application
2. Which of the following OSI model layers includes pass-through services for the session layer functions?
   - **A.** Data-link
   - **B.** Application
   - **C.** Network
   - **D.** Presentation
3. Which of the following is an example of a circuit-switching network?
   - **A.** A local area network running Ethernet
   - **B.** The PSTN
   - **C.** A wireless local area network
   - **D.** A cable television network
4. Which of the following operating system kernels is unable to operate using the peer-to-peer model?
   - **A.** Windows
   - **B.** UNIX
   - **C.** Linux
   - **D.** Novell NetWare

Answers

This section contains the answers to the questions for the Exercise and Chapter Review in this chapter.

Exercise

1. Data-link
2. Transport
3. Network
4. Network
5. Transport
6. Data-link
7. Application
8. Data-link
9. Transport
10. Data-link
11. Session
12. Physical
13. Presentation
14. Network
15. Transport

Chapter Review

1. **Correct Answer:** C
   - A. **Incorrect:** The data-link layer protocol provides error detection, but not error correction.
   - B. **Incorrect:** The network layer protocol provides error correction, but only of the network layer protocol header.
   - C. **Correct:** The transport layer protocol can provide end-to-end error correction for the entire packet.
   - D. **Incorrect:** The application layer protocols do not provide error detection or correction.

2. **Correct Answer:** D

   A. **Incorrect:** Data-link layer protocols do not interact directly with session layer services.

   B. **Incorrect:** The application layer does not require pass-through services, because there are no layers above it.

   C. **Incorrect:** Network layer protocols do not interact directly with session layer services.

   D. **Correct:** The presentation layer includes pass-through services for the 22 functions performed by the session layer, so that application layer protocols can issue calls for session layer services.

3. **Correct Answer:** B

   A. **Incorrect:** All local area networks are packet-switching networks.

   B. **Correct:** The PSTN is a circuit-switching network because the system establishes a connection between two nodes before any user data is transmitted.

   C. **Incorrect:** All local area networks are packet-switching networks.

   D. **Incorrect:** A cable television network is an example of a packet-switching network.

4. **Correct Answer:** D

   A. **Incorrect:** Windows is capable of running in peer-to-peer mode.

   B. **Incorrect:** UNIX operating systems are capable of running in peer-to-peer mode.

   C. **Incorrect:** Linux operating systems are capable of running in peer-to-peer mode.

   D. **Correct:** Novell NetWare can only operate in client/server mode.