Chapter 2

Networking Standards

Before servers can provide services to clients, communications between the two entities must be enabled. Besides the cables that you see, numerous processes operate behind the scenes to keep things running smoothly. For these processes to interoperate smoothly in a diverse networking environment, the computing community has settled on several standards and specifications that define the interaction and interrelation of the various components of network architecture. This chapter explores some of those standards, including the Open Systems Interconnection (OSI) reference model, Serial Line Internet Protocol (SLIP), Point-to-Point Protocol (PPP), the IEEE 802 standards, Network Driver Interface Specification (NDIS), and Open Data-Link Interface (ODI).

Chapter 2 targets the following objectives in the Standards and Terminology section of the Networking Essentials exam:

Test Objectives

- Define the communication devices that communicate at each level of the OSI model
- Compare the implications of using connection-oriented communications with connectionless communications
- Distinguish whether SLIP or PPP is used as the communications protocol for various situations
- Describe the characteristics and purpose of the media used in IEEE 802.3 and IEEE 802.5
- Explain the purpose of the NDIS and Novell ODI network standards

Test Yourself

Stop! Before reading this chapter, test yourself to determine how much study time you will need to devote to this section.

1. Routers operate at the OSI __________ layer.
   - A. Transport
   - B. Network
   - C. Data Link
   - D. Physical

2. __________ communication provides flow control at internal nodes.
   - A. Transport
   - B. Internal
   - C. Connection-oriented
   - D. Internet

3. __________ supports serial line communication under the TCP/IP transport protocol.
   - A. SLIP
   - B. PPP
   - C. Both A and B
   - D. None of the above

4. 10BASE-T networks are defined in the __________ standard.
   - A. IEEE 802.1
   - B. IEEE 802.5
   - C. Both A and B
   - D. None of the above

Standards

The network industry uses two types of standards: *de facto standards* and *de jure standards*. To understand the concept of open systems architecture, you must be familiar with the concepts of de facto and de jure standards.

De facto standards arise through widespread commercial and educational use. These standards often are proprietary and usually remain unpublished and unavailable to outside vendors. Unpublished and unavailable standards are known as *closed system standards*. Published and accessible standards, on the other hand, are known as *open system standards*. Through the introduction of the OSI model, which is discussed later in this chapter, and the growing acceptance of the concept of interoperability, many closed proprietary systems (such as IBM's Systems Network Architecture) have started to migrate toward open system standards. Certainly, de facto standards are not always closed system standards—examples of proprietary open system standards include Novell's NetWare network operating system and Microsoft's Windows.

The second type of standards, de jure standards, are nonproprietary, which means that no single company creates them or owns the rights to them. De jure standards are developed with the intent of enhancing connectivity and interoperability by making specifications public so that independent manufacturers can build to such specifications. TCP/IP is an example of a nonproprietary de jure standard.

Several permanent committees comprised of industry representatives develop de jure standards. Although these committees are supported by manufacturer subscriptions and major company end users, they are intended to represent the interests of the entire community and thus remain independent of any one manufacturer's interests. Subscribing to de jure standards reduces the risk and cost of developing hardware and software for manufacturers. After a standard has been finalized, a component manufacturer subscribing to it can develop products with some confidence that the products will operate with components from other companies.

Standards Organizations and the ISO

The development and implementation of de jure standards is regulated by standards organizations. For example, the International Telecommunication Union (ITU) and the Institute of Electrical and Electronic Engineers (IEEE), among other organizations, are responsible for several prominent network standards that support the International Standards Organization's objective of network interoperability.

The International Standards Organization (ISO)—whose name is derived from the Greek prefix *iso*, meaning "same"—is located in Geneva, Switzerland. ISO develops and publishes standards and coordinates the activities of all national standardization bodies. In 1977, the ISO initiated efforts to design a communication standard based on the open systems architecture theory from which computer networks would be designed. This model came to be known as the Open Systems Interconnection (OSI) model.

Rules and the Communication Process

Networks rely on many rules to manage information interchange. Some of the procedures governed by network standards are as follows:

- Procedures used to initiate and end an interaction
- Signals used to represent data on the media
- Methods used to direct a message to the intended destination
- Procedures used to control the rate of data flow
- Methods used to enable different computer types to communicate
- Ways to ensure that messages are received correctly

Enabling computers to communicate is an extremely complex process—one that is often too complex to solve all at once using just one set of rules. As a result, the industry has chosen to solve different parts of the problem with compatible standards so that the solutions can be put together like pieces of a puzzle—a puzzle that comes together differently each time to build a complete communication approach for any given situation.

The OSI Reference Model

Having a model in mind will help you understand how the pieces of the networking puzzle fit together. The most commonly used model is the Open Systems Interconnection (OSI) reference model. The OSI model, first released in 1984 by the International Standards Organization (ISO), provides a useful structure for defining and describing the various processes underlying open systems networking.

The OSI model is a blueprint for vendors to follow when developing protocol implementations. The OSI model organizes communication protocols into seven layers. Each layer addresses a narrow portion of the communication process. Figure 2.1 illustrates the layers of the OSI model.

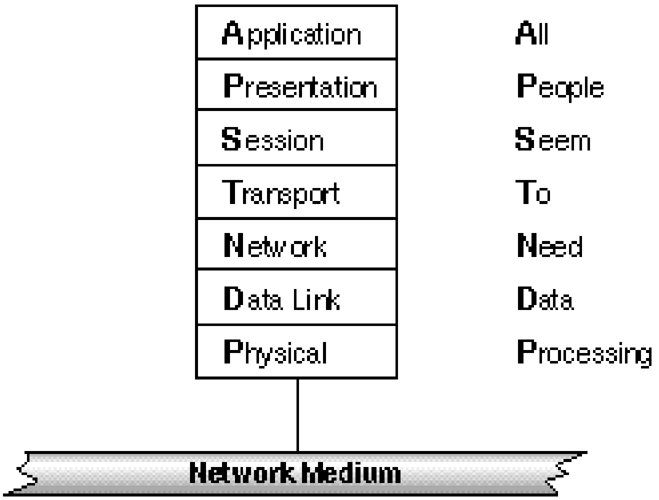


Figure 2.1

The OSI model is composed of seven layers.

Although you will examine each layer in detail later in this chapter, a quick overview is in order. Layer 1, the Physical layer, consists of protocols that control communication on the network media. Layer 7, the Application layer, interfaces the network services with the applications in use on the computer. The five layers in between—Data Link, Network, Transport, Session, and Presentation—perform intermediate communication tasks.

note

> You should learn the names and the order of the seven OSI layers for the Networking Essentials exam. The following two phrases help you remember the first letters of the layers:
>
> **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing (top down)
>
> **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way (bottom up)
>
> Choose one, depending on whether you are most comfortable working from the top of the model down or from the bottom up.

Protocol Stacks

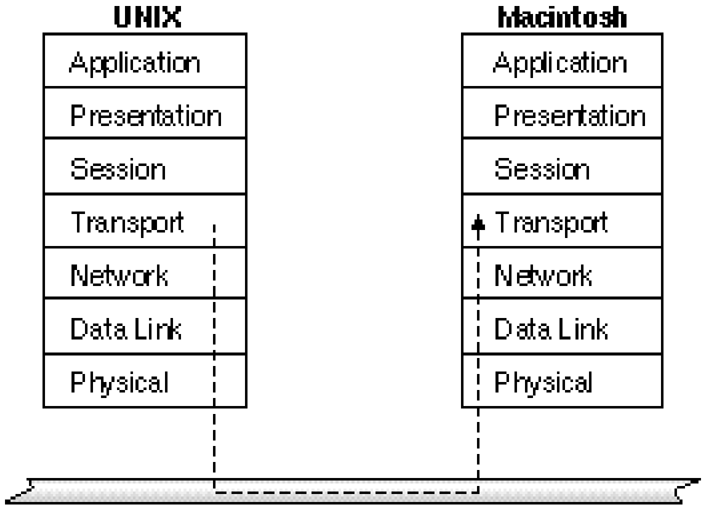
Figure 2.1 illustrates the origin of the term *protocol stack*. The OSI model (and other non-OSI protocol standards) break the complex process of network communication into *layers*. Each layer represents a category of related tasks. A protocol stack is an implementation of this layered protocol architecture. The protocols and services associated with the protocol stack interact to prepare, transmit, and receive network data.

Two computers must run compatible protocol stacks before they can communicate because each layer in one computer's protocol stack must interact with a corresponding layer in the other computer's protocol stack. Figure 2.2, for example, shows the path of a message that starts in the Transport layer. The message travels down the protocol stack, through the network medium, and up the protocol stack of the receiving computer. If the Transport layer in the receiving computer understands the protocols used in the Transport layer that originated the message, the message can be delivered.

As long as their protocol stacks are compatible, computers of different types can communicate. TCP/IP, for example, is available for almost all current computers and operating systems. If a Macintosh and a Unix workstation both run TCP/IP, the Mac can access files on the Unix workstation.

Figure 2.2

Peer communication takes place between protocol stacks.

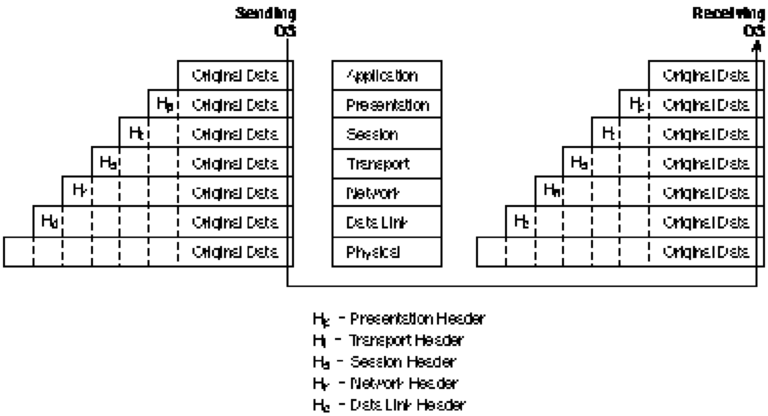


How Peer Layers Communicate

To communicate with its peer layer in another computer, each protocol layer adds its own information to the message being sent. This information takes the form of a *header* added to the beginning of the message (see fig. 2.3).

Figure 2.3

Each protocol layer, except the Physical layer, adds a header to the frame.



Headers are added as the message is prepared for transmission, and headers are removed (stripped) by the receiving computer after the information in the header has been utilized.

note

> The Physical layer does not append a header because this layer deals with sending and receiving information on the individual bit level. The bits are assembled into longer message units in the Data Link layer.

OSI Physical Layer Concepts

Although the OSI Physical layer does not define the media used, this layer is concerned with all aspects of transmitting and receiving data on the network media. Specifically, the Physical layer is concerned with transmitting and receiving bits. This layer defines several key characteristics of the physical network, including the following:

- Physical structure of the network (physical topology)
- Mechanical and electrical specifications for using the medium (not the medium itself)
- Bit transmission encoding and timing

Although the Physical layer does not define the physical medium, it defines clear requirements that the medium must meet. These specifications differ depending on the physical medium. Ethernet for UTP, for example, has different Physical layer specifications from coaxial Ethernet (see Chapter 3, "Transmission Media").

note

> You learn more about network transmission media in Chapter 3. In Chapter 4, "Network Topologies and Architectures," you learn about physical topologies.

OSI Data Link Layer Concepts

As you learned in the preceding section, the OSI Physical layer is concerned with moving messages at the machine level. Network communication, however, is considerably more involved than moving bits from one device to another. In fact, dozens of steps must be performed to transport a message from one device to another.

Real messages consist not of single bits but of meaningful groups of bits. The Data Link layer receives messages, called *frames*, from upper layers. A primary function of the Data Link layer is to disassemble these frames into bits for transmission and then to reconstruct the frames from the bits received.

The Data Link layer has other functions as well, such as addressing, error control, and flow control for a single link between network devices. (The adjacent Network layer, described later in this chapter, handles the more complex tasks associated with addressing and delivering packets through routers and across an internetwork.)

The IEEE 802 standard divides the Data Link layer into two sublayers:

- **Media Access Control (MAC).** The MAC sublayer controls the means by which multiple devices share the same media channel. This includes contention methods (see Chapter 4) and other media access details. The MAC layer also provides addressing information for communication between network devices.
- **Logical Link Control (LLC).** The LLC sublayer establishes and maintains links between communicating devices.

Hardware Access at the Data Link Layer

As the preceding section mentions, the Data Link layer's MAC sublayer provides an interface to the network adapter card. The details necessary to facilitate access to the network through the adapter card are thus assigned to the Data Link layer. Some of these details include the access control method (for example, contention or token passing—see Chapter 4) and the network topology.

The Data Link layer also controls the transmission method (for example, synchronous or asynchronous) used to access the transmission medium. See Chapter 6, "Connectivity Devices," for more on synchronous and asynchronous communications.

Addressing at the Data Link Layer

The Data Link layer maintains device addresses that enable messages to be sent to a particular device. The addresses are called *physical device addresses*. Physical device addresses are unique addresses associated with the networking hardware in the computer. In most cases (for example, Ethernet and Token Ring), the physical device address is burned into the network interface card at the time the card is manufactured.

The standards that apply to a particular network determine the format of the address. Because the address format is associated with the media access control method used, physical device addresses are frequently referred to as *MAC addresses*.

The device address is not actually used to route a message to a specific device. Frames on LANs are typically transmitted so that they are available to all devices on the network. Each device reads each frame far enough to determine the device address to which the frame is addressed. If the frame's destination address matches the device's own physical address, the rest of the frame is received. If the addresses do not match, the remainder of the frame is ignored.

As you learn in this chapter, *bridges* can be used to divide large networks into several smaller ones. Bridges use physical device addresses to determine which frames to leave on the current network segment and which to forward to devices on other network segments. (Chapter 6 covers bridges in more detail.)

Because they use physical device addresses to manage frame routing, bridges function at the level of the Data Link layer and are Data Link layer connectivity devices.

Error and Flow Control at the Data Link Layer

Several of the protocol layers in the OSI model play a role in the overall system of flow control and error control for the network. Flow control and error control are defined as follows:

- **Flow control.** Flow control determines the amount of data that can be transmitted in a given time period. Flow control prevents the transmitting device from overwhelming the receiver.
- **Error control.** Error control detects errors in received frames and requests retransmission of frames.

The Data Link layer's LLC sublayer provides error control and flow control for single links between communicating devices. The Network layer (described in the following section) expands the system of error control and flow control to encompass complex connections that include routers, gateways, and internetworks.

OSI Network Layer Concepts

As you learned in the preceding section, the Data Link layer deals with communication between devices on the same network. Physical device addresses are used to identify data frames, and each device is responsible for monitoring the network and receiving frames addressed to that device.

The Network layer handles communication with devices on logically separate networks that are connected to form *internetworks*. Because internetworks can be large and can be constructed of different types of networks, the Network layer utilizes routing algorithms that guide packets from their source to their destination networks. For more about routing and routing algorithms, see Chapter 6.

Within the Network layer, each network in the internetwork is assigned a *network address* that is used to route packets. The Network layer manages the process of addressing and delivering packets on complex networks.

Network Layer Addressing

You have already encountered the Data Link layer's physical device addresses, which uniquely identify each device on a network. On larger networks, it is impractical to deliver network data soley by means of physical addresses. (Imagine if your network adapter had to check every packet sent from anywhere on the Internet to look for a matching physical address.) Larger networks require a means of routing and filtering packets in order to reduce network traffic and minimize transmission time. The Network layer uses *logical network addresses* to route packets to specific networks on an internetwork.

Logical network addresses are assigned during configuration of the networks. A network installer must make sure that each network address is unique on a given internetwork.

The Network layer also supports *service addresses.* A service address specifies a channel to a specific process on the destination PC. The operating systems on most computers can run several processes at once. When a packet arrives, you must determine which process on the computer should receive the data in the packet. You do so by assigning service addresses, which identify upper-layer processes and protocols. These service addresses are included with the physical and logical network addresses in the data frame. (Some protocols refer to service addresses as *sockets* or *ports.*)

Some service addresses, called *well-known addresses*, are universally defined for a given type of network. Other service addresses are defined by the vendors of network products.

Delivering Packets

Many internetworks often include redundant data paths that you can use to route messages. Typically, a packet will pass from the local LAN segment of the source PC through a series of routers to the local LAN segment of the destination PC. (You learn more about routers in Chapter 6.) The OSI Network layer oversees the process of determining paths and delivering packets across the internetwork. Chapter 6 describes some of the routing algorithms used to determine a path. The following sections introduce some of the basic switching techniques, as follows:

- Circuit switching
- Message switching
- Packet switching

Circuit Switching

Circuit switching establishes a path that remains fixed for the duration of a connection (see fig. 2.4). Much as telephone switching equipment establishes a route between two telephones, circuit-switching networks establish a path through the internetwork when the devices initiate a conversation.

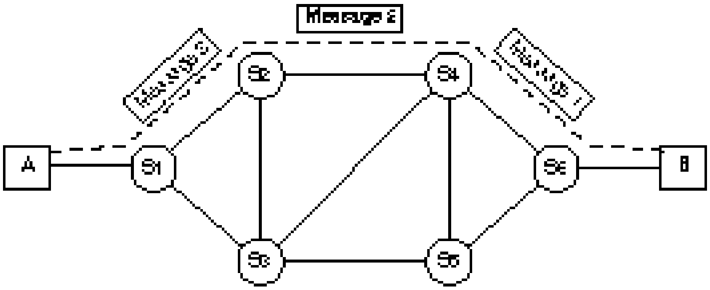


Figure 2.4

Circuit switching establishes a path between two devices much like a telephone connection.

Circuit switching provides devices with a dedicated path and a well-defined bandwidth, but circuit switching is not free of disadvantages. First, establishing a connection between devices can be time-consuming. Second, because other traffic cannot share the dedicated media path, bandwidth might be inefficiently utilized. Finally, circuit-switching networks must have a surplus of bandwidth, so these types of switches tend to be expensive to construct.

Message Switching

Message switching treats each message as an independent entity. Each message carries address information that describes the message's destination, and this information is used at each switch to transfer the message to the next switch in the route. Message switches are programmed with information concerning other switches in the network that can be used to forward messages to their destinations. Message switches also may be programmed with information about the most efficient routes. Depending on network conditions, different messages may be sent through the network by different routes, as shown in figure 2.5.

Message switching transfers the complete message from one switch to the next, where the message is stored before being forwarded again. Because each message is stored before being sent on to the next switch, this type of network frequently is called a *store-and-forward network*. The message switches often are general-purpose computers and must be equipped with sufficient storage (usually hard drives) to enable them to store messages until forwarding is possible.

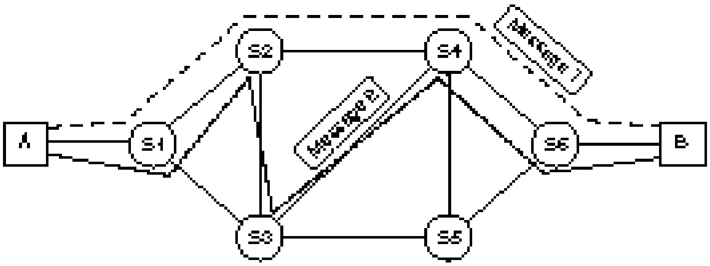


Figure 2.5

Message switching forwards the complete message one switch at a time.

Message switching commonly is used in e-mail because some delay is permissible when delivering mail, unlike the requirements when two computers exchange data in real time. Message switching uses relatively low-cost devices to forward messages and can function well with relatively slow communication channels. Other applications for message switching include group applications such as workflow, calendaring, and groupware.

Message switching offers the following advantages:

- Data channels are shared among communicating devices, improving the efficiency of using available bandwidth.
- Message switches can store messages until a channel becomes available, reducing sensitivity to network congestion.
- Message priorities can be used to manage network traffic.
- Broadcast addressing uses network bandwidth more efficiently by delivering messages to multiple destinations.

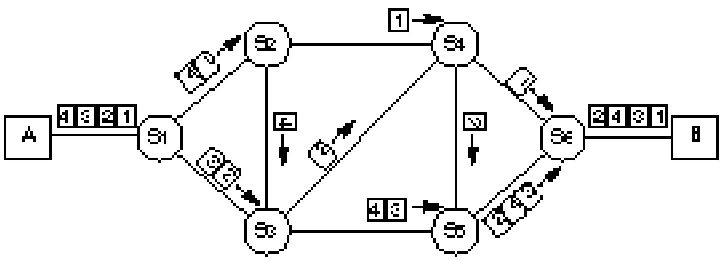
The chief disadvantage of message switching is that message switching is not suited for real-time applications, including data communication, video, and audio.

Packet Switching

In packet switching, messages are divided into smaller pieces called *packets*. Each packet includes source and destination address information so that individual packets can be routed through the internetwork independently. As you can see in figure 2.6, the packets that make up a message can take very different routes through the internetwork.

Figure 2.6

Packet switching breaks up the message into smaller pieces that are routed independently.



So far, packet switching looks considerably like message switching, but the distinguishing characteristic is that packets are restricted to a size that enables the switching devices to manage the packet data entirely in memory. This eliminates the need to store the data temporarily on disk. Packet switching, therefore, routes packets through the network much more rapidly and efficiently than is possible with message switching.

Several methods of packet switching exist. Two common methods of packet switching are as follows:

- Datagram
- Virtual circuit

These two methods are discussed in the following sections.

Datagram Packet Switching

Datagram services treat each packet as an independent message. Each packet is routed through the internetwork independently, and each switch node determines which network segment should be used for the next step in the packet's route. This capability enables switches to bypass busy segments and take other steps to speed packets through the internetwork (refer to fig. 2.6).

Datagrams are frequently used on LANs. Network layer protocols are responsible for delivering the frame to the appropriate network. Then, because each datagram includes destination address information, devices on the local network can recognize and receive appropriate datagrams.

note

> Packet switching meets the need to transmit large messages with the fairly small frame size that can be accommodated by the Physical layer. The Network layer is responsible for fragmenting messages from upper layers into smaller datagrams that are appropriate for the Physical layer. The Network layer also is responsible for reconstructing messages from datagrams as they are received.

Virtual Circuit Packet Switching

Virtual circuits operate by establishing a formal connection between two devices in communication. When devices begin a session, they negotiate communication parameters, such as maximum message size, communication windows, and network paths. This negotiation establishes a *virtual circuit,* which is a well-defined path through the internetwork by which the devices communicate. This virtual circuit generally remains in effect until the devices stop communicating.

Virtual circuits are distinguished by the establishment of a logical connection. *Virtual* means that the network behaves as though a dedicated physical circuit has been established between the communicating devices. Even though no such physical circuit actually exists, the network presents the appearance of a physical connection to the devices at the ends of the circuit.

Virtual circuits frequently are employed in conjunction with connection-oriented services, which are discussed later in this chapter.

Advantages of Packet Switching

Packet switching optimizes the use of bandwidth by enabling many devices to route packets through the same network channels. At any given time, a switch can route packets to several different destination devices, adjusting the routes as required to achieve the best efficiency.

Because entire messages are not stored at the switches prior to forwarding, transmission delays are significantly shorter than delays encountered with message switching.

Although the switching devices do not need to be equipped with large amounts of hard drive capacity, they might need a significant amount of real-time memory. In addition, the switching devices must have sufficient processing power to run the more complex routing protocols required for packet switching. A system must be in place by which devices can recognize when packets have been lost so that retransmission can be requested.

Connection-oriented and Connectionless Modes

The OSI Network layer determines the route a packet will take as it passes through a series of routers from the source PC to the destination PC. The complexity and versatility of Network layer addressing gives rise to two different communication modes for passing messages across the network, both of which are recognized under OSI:

- **Connection-oriented mode.** Error correction and flow control are provided at internal nodes along the message path.
- **Connectionless mode.** Internal nodes along the message path do not participate in error correction and flow control.

To understand the distinction between connection-oriented and connectionless communications, you must consider an important distinction between the OSI model's Data Link and Network layers. In theory, the Data Link layer facilitates the transmission of data across a single link between two nodes. The Network layer describes the process of routing a packet through a series of nodes to a destination elsewhere on the network. An example of this latter scenario is a message passing from a PC on one LAN segment through a series of routers to a PC on a distant part of the network. The internal nodes forwarding the packet also forward other packets between other end nodes.

In connection-oriented mode, the chain of links between the source and destination nodes forms a kind of logical pathway—a connection. The nodes forwarding the data packet can track which packet is part of which connection. This enables the internal nodes to provide flow control as the data moves along the path. For example, if an internal node determines that a link is malfunctioning, the node can send a notification message backwards through the path to the source computer. Furthermore, because the internal node distinguishes among individual, concurrent connections in which it participates, this node can transmit (or forward) a "stop sending" message for one of its connections without stopping all communications through the node. Another feature of connection-oriented communication is that internal nodes provide error correction at each link in the chain. Therefore, if a node detects an error, it asks the preceding node to retransmit.

Connectionless mode does not provide these elaborate internal control mechanisms; instead, connectionless mode relegates all error-correcting and retransmitting processes to the source and destination nodes. The end nodes acknowledge the receipt of packets and retransmit if necessary, but internal nodes do not participate in flow control and error correction (other than simply forwarding messages between the end nodes).

The advantage of connectionless mode is that connectionless communications can be processed more quickly and more simply because the internal nodes only forward data and thus don't have to track connections or provide retransmission or flow control.

Connectionless mode does have its share of disadvantages, however, including the following:

- Messages sometimes get lost due to an overflowing buffer or a failed link along the pathway.
- If a message gets lost, the sender doesn't receive notification.
- Retransmission for error correction takes longer because a faulty transmission can't be corrected across an internal link.

It is important to remember that OSI is not a protocol suite itself—OSI is a standard for designing protocol suites. As such, individual implementations of connectionless protocols can attenuate some of the preceding disadvantages. It is also important to remember that connection-oriented mode, although it places much more emphasis on monitoring errors and controlling traffic, doesn't always work either. Ultimately, the choice of connection-oriented or connectionless communications mode depends on interoperability with other systems, the premium for speed, and the cost of components.

Gateway Services

Routers can handle interconnection of networks whose protocols function in similar ways. When the rules differ sufficiently on the two networks, however, a more powerful device is required.

A *gateway* is a device that can reconcile the different rules used on two different networks. Gateways commonly are required to connect LANs to mainframe networks, which have completely different protocol architectures. Mainframe networks, such as IBM's SNA, for example, do not use the same device address schemes that LANs employ (these networks differ in many other ways as well). In these situations, you must fool the mainframe network into thinking that mainframe devices are on the LAN. This involves making the mainframe look like a LAN to devices on the LAN. Exercise 2.1, at the end of this chapter, explores Windows NT's Gateway Services for NetWare, which provides a gateway from a Microsoft network to NetWare resources.

note

> Gateways can be implemented at the Network layer or at higher layers in the OSI model, depending on where the protocol translation is required.

OSI Transport Layer Concepts

The Transport layer, the next layer of the OSI model, can implement procedures to ensure the reliable delivery of messages to their destination devices. The term "reliable" does not mean that errors cannot occur; instead, it means that if errors occur, they are detected. If errors such as lost data are detected, the Transport layer either requests retransmission or notifies upper-layer protocols so that they can take corrective action.

The Transport layer enables upper-layer protocols to interface with the network but hides the complexities of network operation from them. Among its functions, the Transport layer breaks large messages into segments suitable for network delivery.

Transport Layer Connection Services

Some services can be performed at more than one layer of the OSI model. In addition to the Data Link and Network layers, the Transport layer can take on some responsibility for connection services. The Transport layer interacts with the Network layer's connection-oriented and connectionless services and provides some of the essential quality control features. Some of the Transport layers activities include the following:

- **Segment sequencing.** Segment sequencing is one connection-oriented service provided by the Transport layer. When large messages are divided into segments for transport, the Transport layer must resequence the segments when they are received before reassembling the original message.

- **Error control.** When segments are lost during transmission or when segments have duplicate segment IDs, the Transport layer must initiate error recovery. The Transport layer also detects corrupted segments by managing end-to-end error control using techniques such as checksums.

- **End-to-end flow control.** The Transport layer uses acknowledgments to manage end-to-end flow control between two connected devices. Besides negative acknowledgments, some Transport-layer protocols can request the retransmission of the most recent segments.

OSI Session Layer Concepts

The next OSI layer, the Session layer, manages dialogs between two computers by establishing, managing, and terminating communications. As illustrated in figure 2.7, dialogs can take three forms:

- **Simplex dialogs.** These dialogs are responsible for only one-way data transfers. An example is a fire alarm, which sends an alarm message to the fire station but cannot (and does not need to) receive messages from the fire station.

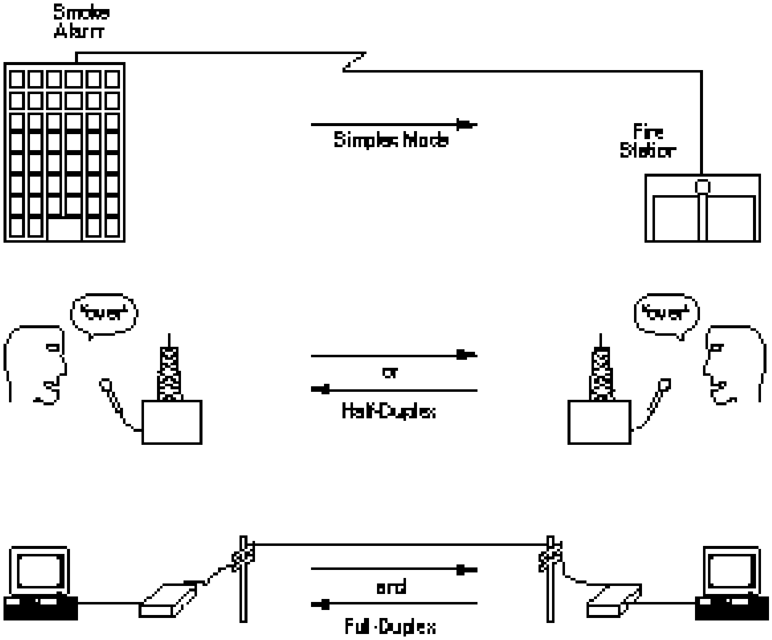
- **Half-duplex dialogs.** These dialogs handle two-way data transfers in which the data flows in only one direction at a time. When one device completes a transmission, this device must "turn over" the medium to the other device so that this second device has a turn to transmit.

  CB radio operators, for example, converse on the same communication channel. When one operator is finished transmitting, he must release his transmit key so that the other operator can send a response.

- **Full-duplex dialogs.** This third type of dialog permits two-way simultaneous data transfers by providing each device with a separate communication channel. Voice telephones are full-duplex devices, and either party to a conversation can talk at any time. Most computer modems can operate in full-duplex mode.

Figure 2.7

Simplex and duplex communication modes.



Costs rise for half- and full-duplex operation because the more complex dialog technologies are naturally more expensive. Designers of communications systems, therefore, generally use the simplest dialog mode that satisfies the communication requirements.

Half-duplex communication can result in wasted bandwidth during the intervals when communication is turned around. On the other hand, using full-duplex communication generally requires a greater bandwidth than half-duplex communication.

The Session layer also marks the data stream with checkpoints and monitors the receipt of those checkpoints. In the event of a failure, the sending PC can retransmit starting with the data sent after the last checkpoint, rather than resending the whole message.

Session Layer Session Administration

A *session* is a formal dialog between a service requester and a service provider. Sessions have at least three phases:

- **Connection establishment.** In this phase, a service requester requests initiation of a service. During the setup process, communication is established and rules are agreed upon.
- **Data transfer.** Because of the rules agreed upon during setup, each party to the dialog knows what to expect. Communication is therefore efficient, and errors are easy to detect.
- **Connection release.** When the session is completed, the dialog is terminated in an orderly fashion.

The connection establishment phase establishes the parameters for the communication session. Actually, the connection establishment phase is comprised of several tasks, including the following:

- Specification of required services
- User login authentication and other security procedures
- Negotiation of protocols and protocol parameters
- Notification of connection IDs
- Establishment of dialog control, as well as acknowledgment of numbering and retransmission procedures

After the connection is established, the devices involved can initiate a dialog (data transfer phase). Besides exchanging data, these devices exchange acknowledgments and other control data that manage the dialog.

The Session layer also can incorporate protocols to resume dialogs that have been interrupted. After a formal dialog has been established, devices recognize a lost connection whenever the connection has not been formally released. Therefore, a device realizes that a connection has been lost when the device fails to receive an expected acknowledgment or data transmission.

Within a certain time period, two devices can reenter the session that was interrupted but not released. The connection release phase is an orderly process that shuts down communication and releases resources on the service provider.

OSI Presentation Layer Concepts

The Presentation layer deals with the syntax, or grammatical rules, needed for communication between two computers. The Presentation layer converts system-specific data from the Application layer into a common, machine-independent format that will support a more standardized design for lower protocol layers.

The Presentation layer also attends to other details of data formatting, such as data encryption and data compression.

note

> The name "Presentation layer" has caused considerable confusion in the industry because some people mistakenly believe that this layer presents data to the user. However, the name has nothing to do with displaying data. Instead, this function is performed by applications running above the Application layer.
>
> The Presentation layer is so named because it presents a uniform data format to the Application layer. As a matter of fact, this layer is not commonly implemented because applications typically perform most Presentation layer functions.

On the receiving end, the Presentation layer converts the machine-independent data from the network into the format required for the local system. This conversion could include the following:

- **Bit-order translation.** When binary numbers are transmitted through a network, they are sent one bit at a time. The transmitting computer can start at either end of the number. Some computers start at the *most-significant digit (MSD);* others start at the *least-significant digit (LSD).*
- **Byte-order translation.** Complex values generally must be represented with more than one byte, but different computers use different conventions to determine which byte should be transmitted first. Intel microprocessors, for example, start with the least-significant byte and are called *little*

endian. Motorola microprocessors, on the other hand, start with the most-significant byte and are called *big endian*. Byte-order translation might be needed to reconcile these differences when transferring data between a PC and a Macintosh.

- **Character code translation.** Different computers use different binary schemes for representing character sets. For instance: *ASCII,* the American Standard Code for Information Interchange, is used to represent English characters on all microcomputers and most minicomputers (see fig. 2.8); *EBCDIC,* the Extended Binary Coded Decimal Interchange Code, is used to represent English characters on IBM mainframes (see fig. 2.9); and *Shift-JIS* is used to represent Japanese characters.
- **File Syntax Translation.** File formats differ between computers. For instance, Macintosh files actually consist of two related files called a data fork and a resource fork. PC files, on the other hand, consist of a single file.

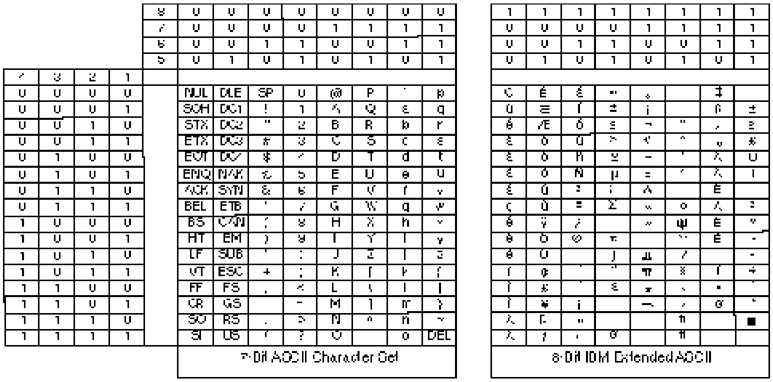
note

> Many vendors are beginning to incorporate *Unicode* in their products. Unicode, a 16-bit code that can represent 65,536 characters in English and other languages, is organized into code pages devoted to the characters required for a given language. Unicode improves the portability of products between different language environments.

The redirector service (see Chapter 1, "Networking Terms and Concepts") operates at the OSI Presentation layer.

Figure 2.8

The ASCII character code represents English characters on microcomputers and most minicomputers.



4	3	2	1	8: 0 7: 0 6: 0 5: 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

7-Bit ASCII Character Set

8: 1 7: 0 6: 0 5: 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
Ç	É	á	∞	◊		‡	
ü	æ	í	±	¡		ß	±
é	Æ	ó	≤	¬	"	‚	≥
â	ô	ú	≥	√	^	µ	*
ä	ö	ñ	¥	–	'	Λ	U
à	ò	Ñ	µ	=	´	Λ	I
å	û	ª	∂	∧		È	
ç	ù	º	Σ	«	○	Λ	²
ê	ÿ	¿		»	ψ	È	ᵛ
ë	Ö	⌐	π		¨	È	-
è	Ü		∫	╨	⁄		-
ï	¢	'	"	╥	⁸	Í	÷
î	£	'	ε	╤	‛	▪	˙
ì	¥	¡		─	⁄	σ	˚
Ä	₧	"			ħ		■
Å	ƒ	'	σ		ħ		

8-Bit IBM Extended ASCII

Figure 2.9

The EBCDIC character code represents English characters on IBM mainframes.

OSI Application Layer Concepts

The Application layer of the OSI reference model is concerned with providing services on the network, including file services, print services, e-mail services, and database services, among others.

A common misunderstanding is that the Application layer is responsible for running user applications such as word processors. This is not the case. The Application layer, however, does provide an interface whereby applications can communicate with the network.

The Application layer also advertises the available services to the network.

Communications Devices and OSI

Each of the three lowest levels of the OSI model supports an important network hardware device. The function of these devices (and the differences between them) sheds some light on the functions of their corresponding OSI layers. This section describes how the following devices fit within the OSI model:

- **Repeaters.** Operate at the OSI Physical layer
- **Bridges.** Operate at the OSI Data Link layer
- **Routers (and brouters).** Operate at the OSI Network layer

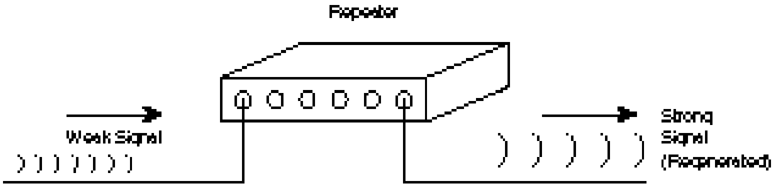
These devices are discussed in greater detail in Chapter 6. The *network adapter* (an all-important connectivity component that operates at the OSI Data Link layer) is discussed in Chapter 10, "Network Adapter Cards."

Repeaters

A *repeater* is a network device that repeats a signal from one port onto the other ports to which it is connected (see fig. 2.10). Repeaters operate at the OSI Physical layer. A repeater does not filter or interpret anything; instead, it merely repeats (regenerates) a signal, passing all network traffic in all directions.

Figure 2.10

A repeater regenerates a weak signal.



A repeater can operate at the OSI Physical layer because a repeater doesn't require any addressing information from the data frame. Therefore, the repeater doesn't have to pass the frame to upper layers where addresses and other parameters are interpreted. A repeater merely passes along bits of data, even if a data frame is corrupt. In fact, a repeater even will forward a broadcast storm caused by a malfunctioning adapter (see Chapter 13, "Troubleshooting," for details).

The primary purpose of a repeater is to enable the network to expand beyond the distance limitations of the transmission medium (see Chapter 3 for more details).

The advantages of repeaters are that they are inexpensive and simple. In addition, though they cannot connect networks with dissimilar data frames (such as a Token Ring network to an Ethernet network), some repeaters can connect segments with similar frame types but dissimilar cabling.

Bridges

A *bridge* is a connectivity device that operates at the OSI Data Link layer. The messaging parameters available at the Data Link layer enable a bridge to pass a frame in the direction of its destination without simultaneously forwarding it to segments for which it was not intended. In other words, a bridge can filter network traffic. This filtering process reduces overall traffic because the bridge segments the network, passing frames only when they can't be delivered on the local segment and passing frames only to the segment for which they are intended.

Figure 2.11 depicts a simple bridge implementation. In this process, a bridge filters traffic by tracking and checking the Data Link layer's MAC sublayer addresses of incoming frames. The bridge monitors the source addresses of incoming frames and builds a routing table that shows which nodes are on each of the segments. When a data frame arrives, the bridge checks the frame's destination address and forwards the frame to the segment that contains the destination node. If the destination node exists on the same segment as the source node, the bridge stops the frame so it doesn't pass unnecessarily to the rest of the network. If the bridge can't find the destination address in its routing table, it forwards the frame to all segments except the source segment.

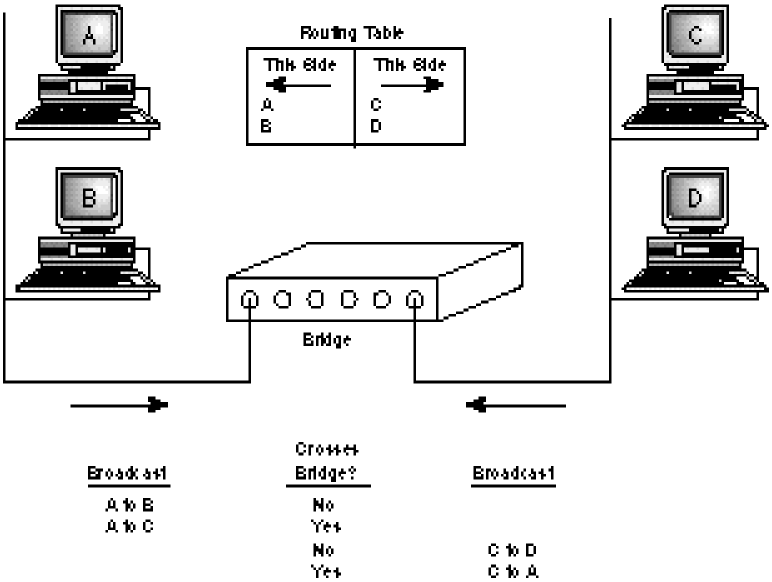


Figure 2.11

A simple bridge configuration.

A bridge can perform the same functions a repeater performs, including expanding cabling distance and linking dissimilar cable types. In addition, a bridge can improve performance and reduce network traffic by splitting the network and confining traffic to smaller segments.

Routers

A *router* is a connectivity device that operates at the OSI Network layer. The information available at the Network layer gives a router far more sophisticated packet-delivery capabilities than a bridge provides. As with a bridge, a router constructs a routing table, but the Network layer addressing information (discussed earlier in this chapter) enables routers to pass packets through a chain of other routers, or even choose the best route for a packet if several routes exist. (See Chapter 6 for more information on routers and how they operate.)

A hybrid device called a *brouter* combines some characteristics of a router and a bridge. A brouter routes routable protocols using information available at the Network layer and acts as a bridge for nonroutable protocols. (A routable protocol is a protocol that can pass through a router. TCP/IP and IPX/SPX are examples of routable protocols—see Chapter 5, “Transport Protocols,” for more information.)

Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP)

Two other standards vital to network communication are *Serial Line Internet Protocol (SLIP)* and *Point-to-Point Protocol (PPP)*. SLIP and PPP were designed to support dial-up access to networks based on the Internet protocols. SLIP is a simple protocol that functions at the Physical layer, whereas PPP is a considerably enhanced protocol that provides Physical layer and Data Link layer functionality. The relationship of both to the OSI model is shown in figure 2.12.

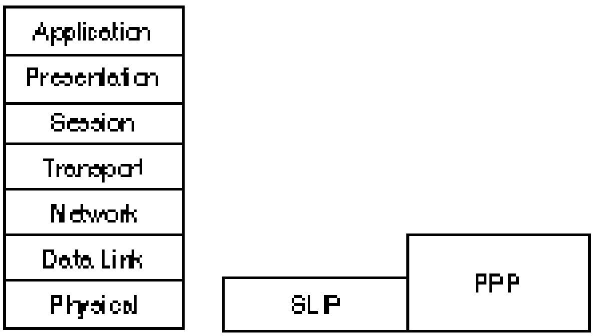


Figure 2.12

The relationship of SLIP and PPP to the OSI reference model.

Developed to provide dial-up TCP/IP connections, SLIP is an extremely rudimentary protocol that suffers from a lack of rigid standardization, which sometimes hinders different implementations from interoperating.

Windows NT supports both SLIP and PPP from the client end using the Dial-Up Networking application. On the server end, Windows NT RAS (Remote Access Service) supports PPP but doesn't support SLIP. In other words, Windows NT can act as a PPP server but not as a SLIP server.

SLIP is most commonly used on older systems or for dial-up connections to the Internet via SLIP-server Internet hosts.

PPP was defined by the Internet Engineering Task Force (IETF) to improve on SLIP by providing the following features:

- Security using password logon
- Simultaneous support for multiple protocols on the same link
- Dynamic IP addressing
- Error control

Different PPP implementations might offer different levels of service and negotiate service levels when connections are made. Because of its versatility, interoperability, and additional features, PPP is presently surpassing SLIP as the most popular serial-line protocol.

Certain dial-up configurations cannot use SLIP for the following reasons:

- **SLIP supports the TCP/IP transport protocol only.** PPP, however, supports TCP/IP, as well as a number of other protocols, such as NetBEUI, IPX, AppleTalk, and DECnet. In addition, PPP can support multiple protocols over the same link.

- **SLIP requires static IP addresses.** Because SLIP requires static IP addresses, SLIP servers do not support Dynamic Host Configuration Protocol (DHCP), which assigns IP addresses dynamically. (DHCP enables clients to share IP addresses so that a relatively small number of IP addresses can serve a larger user base.) If the dial-up server uses DHCP to assign an IP address to the client, the dial-up connection won't use SLIP.

- **SLIP does not support dynamic addressing through DHCP.** Because of this, SLIP connections cannot dynamically assign a WINS or DNS server. (See exercise 2.2 at the end of this chapter.)

note

Windows NT RAS (using PPP) offers a number of other interesting features, including the following:

- **PPP Multilink Protocol.** Multilink enables a single connection to use several physical pathways (such as modems, ISDN lines, and X.25 cards). Utilizing multiple pathways for a single connection increases bandwidth and, therefore, performance.

- **NetBIOS Gateway.** A RAS server can connect a client running the NetBEUI protocol with a TCP/IP or IPX network by serving as a NetBIOS gateway.

- **IPX or IP Router.** A RAS server can act as a router for IPX/SPX and TCP/IP networks. (See Chapter 6 for more information on routers.)

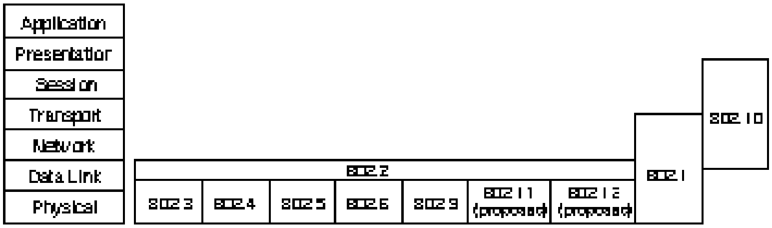
The IEEE 802 Family

The Institute of Electrical and Electronic Engineers (IEEE) is the largest professional organization in the world and is extremely influential with regard to setting standards. The 802 committee of the IEEE, for example, has developed a series of standards for LANs and WANs. The IEEE standards govern lower-layer protocols and interactions with transmission media. These standards have been recognized and reissued by the ISO as the ISO 802 standards.

Twelve subcommittees oversee the 802 standards. (A thirteenth committee has been proposed for the development of the 100BASE-X standard.) Figure 2.13 illustrates the position each standard occupies in the OSI reference model.

Figure 2.13

The relationship between the IEEE 802 standards and the OSI reference model.



IEEE 802.2

The IEEE 802.2 standard defines an LLC sublayer that is used by other lower-layer protocols. Because these lower-layer protocols can use a single LLC protocol layer, Network layer protocols can be designed independently of both the network's Physical layer and MAC sublayer implementations.

The LLC appends to packets a header that identifies the upper-layer protocols associated with the frame. This header also declares the processes that are the source and destination of each packet.

IEEE 802.3

The IEEE 802.3 standard defines a network derived from the Ethernet network originally developed by Digital, Intel, and Xerox. This standard defines characteristics related to the MAC sublayer of the Data Link layer and the OSI Physical layer. With one minor distinction—frame type—IEEE 802.3 Ethernet functions identically to DIX Ethernet v.2. These two standards can even coexist on the same cabling system, although devices using one standard cannot communicate directly with devices using the other.

The MAC sublayer uses a type of contention access called *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*. This technique reduces the incidence of collision by having each device listen to the network to determine whether it's quiet ("carrier sensing"); a device attempts to transmit only when the network is quiescent. This reduces but does not eliminate collisions because signals take some time to propagate through the network. As devices transmit, they continue to listen so they can detect a collision should it occur. When a collision occurs, all devices cease transmitting and send a "jamming" signal that notifies all stations of the collision. Then, each device waits a random amount of time before attempting to transmit again. This combination of safeguards significantly reduces collisions on all but the busiest networks.

IEEE 802.4

The 802.4 standard describes a network with a bus physical topology that controls media access with a token mechanism. This standard was designed to meet the needs of industrial automation systems but has gained little popularity. Both baseband and broadband (using 75-ohm coaxial cable) configurations are available.

IEEE 802.5

The IEEE 802.5 standard was derived from IBM's Token Ring network, which employs a ring logical topology and token-based media-access control. Data rates of 1, 4, and 16 Mbps have been defined for this standard.

IEEE 802.6

The IEEE 802.6 standard describes a MAN standard called *Distributed Queue Dual Bus (DQDB)*. Much more than a data network technology, DQDB is suited to data, voice, and video transmissions. The network is based on fiber-optic cable in a dual-bus topology, and traffic on each bus is unidirectional. When operated in pairs, the two buses provide a fault-tolerant configuration. Bandwidth is allocated using time slots, and both synchronous and asynchronous modes are supported.

IEEE 802.9

The IEEE 802.9 standard supports a 10-Mbps asynchronous channel, along with 96 64-Kbps (6 Mbps total bandwidth) of channels that can be dedicated to specific data streams. The total bandwidth is 16 Mbps. This standard is called *Isochronous Ethernet (IsoEnet)* and is designed for settings with a mix of bursty and time-critical traffic.

IEEE 802.11

IEEE 802.11 is a standard for wireless LANs and is currently under development. A CSMA/CD method has been approved, but the final standard is pending.

IEEE 802.12

The IEEE 802.12 standard is based on a 100-Mbps proposal promoted by AT&T, IBM, and Hewlett-Packard. Called *100VG-AnyLAN*, the network is based on a star-wiring topology and a contention-based access method whereby devices signal the wiring hub when they need to transmit data. Devices can transmit only when granted permission by the hub. This standard is intended to provide a high-speed network that can operate in mixed Ethernet and Token Ring environments by supporting both frame types.

IEEE 802.3 and IEEE 802.5 Media

IEEE 802.2(topology independent), IEEE 802.3 (based on Ethernet), and IEEE 802.5 (based on Token Ring) are the most commonly used IEEE 802 standards. Carefully read the following overview of the media each uses—Microsoft expects you to describe "the characteristics and purpose of the media used in IEEE 802.3 and IEEE 802.5" for the Networking Essentials exam. (Chapters 3 and 4 discuss Ethernet and Token Ring media in greater detail.)

The IEEE 802.3 Physical layer definition describes signaling methods (both baseband and broadband), data rates, media, and topologies. Several Physical layer variants also have been defined.

Each variant is named following a convention that states the signaling rate (1 or 10) in Mbps, baseband (BASE) or broadband (BROAD) mode, and a designation of the media characteristics.

The following list details the IEEE 802.3 variants:

- **1BASE5.** This 1-Mbps network utilizes UTP cable with a signal range up to 500 meters (250 meters per segment). A star physical topology is used.
- **10BASE5.** Typically called Thick Ethernet, or Thicknet, this variant uses a large diameter (10 mm) "thick" coaxial cable with a 50-ohm impedance. A data rate of 10 Mbps is supported with a signaling range of 500 meters per cable segment on a physical bus topology.
- **10BASE2.** Similar to Thicknet, this variant uses a thinner coaxial cable that can support cable runs of 185 meters. (In this case, the "2" only indicates an approximate cable range.) The transmission rate remains at 10 Mbps, and the physical topology is a bus. This variant typically is called Thin Ethernet, or Thinnet.
- **10BASE-F.** This variant uses fiber-optic cables to support 10-Mbps signaling with a range of 4 kilometers. Three subcategories include *10BASE-FL* (fiber link), *10BASE-FB* (fiber backbone), and *10BASE-FP* (fiber passive).
- **10BROAD36.** This broadband standard supports channel signal rates of 10 Mbps. A 75-ohm coaxial cable supports cable runs of 1,800 meters (up to 3,600 meters in a dual-cable configuration) using a physical bus topology.
- **10BASE-T.** This variant uses UTP cable in a star physical topology. The signaling rate remains at 10 Mbps, and devices can be up to 100 meters from a wiring hub.
- **100BASE-X.** This proposed standard is similar to 10BASE-T but supports 100 Mbps data rates.

note

> Some disagreement exists in the industry regarding the proper use of the name “Ethernet.” Xerox has placed the name “Ethernet” in the public domain, which means that no one can claim authority over it. Purists, however, often claim that “Ethernet” refers only to the original Digital-Intel-Xerox standard. More frequently, however, the term designates any network based on CSMA/CD access-control methods.
>
> Usually, it is necessary to be specific about the standard that applies to a given network configuration. The original standard is called Ethernet version 2 (the older version 1 is still in occasional use) or Ethernet-II. The IEEE standard is distinguished by its committee title as 802.3.
>
> This distinction is important because Ethernet version 2 and 802.3 Ethernet use incompatible frame types. Devices using one frame type cannot communicate with devices using the other frame type.

The IEEE 802.5 standard does not describe a cabling system. Most implementations are based on the IBM cabling system, which uses twisted-pair cable wired in a physical star. See Chapter 3 and Chapter 4 for more information on Token Ring cabling and topologies.

NDIS and ODI

The *Network Driver Interface Specification (NDIS)*, a standard developed by Microsoft and 3Com Corp., describes the interface between the network transport protocol and the Data Link layer network adapter driver. The following list details the goals of NDIS:

- To provide a vendor-neutral boundary between the protocol and the network adapter driver so that any NDIS-compliant protocol stack can operate with any NDIS-compliant adapter driver.

- To define a method for binding multiple protocols to a single driver so that the adapter can simultaneously support communications under multiple protocols. In addition, the method enables you to bind one protocol to more than one adapter.

The *Open Data-Link Interface (ODI),* developed by Apple and Novell, serves the same function as NDIS. Originally, ODI was written for NetWare and Macintosh environments. Like NDIS, ODI provides rules that establish a vendor-neutral interface between the protocol stack and the adapter driver. This interface also enables one or more network drivers to support one or more protocol stacks.

Summary

This chapter discussed some of the important standards that define the networking environment. An understanding of these standards is essential for understanding the networking topics discussed in later chapters. This chapter covered the following:

- The OSI model
- SLIP and PPP
- The IEEE 802 Standards
- NDIS and ODI

Later chapters in this book look more closely at related topics, including transmission media, topologies, and connectivity devices.

Exercises

Exercise 2.1: Gateway Services for NetWare

Objective: Install and explore Windows NT Server's Gateway Services for NetWare, an example of a gateway service.

Estimated time: 15 minutes

Gateway Services for NetWare is a Windows NT service that enables a Windows NT Server system to act as a gateway to NetWare resources. A gateway (as described in this chapter) provides a conduit by which computers in one operating environment can access resources located in a dissimilar operating environment. A gateway works by stripping the incompatible protocol layers of an incoming packet and replacing them with the alternative headers needed for the packet to reach its destination. (See Chapter 6 for more information on gateways.) Gateway Services for NetWare provides a Microsoft Network with access to NetWare resources. To a Microsoft Network client that uses Server Message Block (such as Windows NT, Windows 95, and Windows for Workgroups), the NetWare resources will appear as Windows NT resources on the gateway machine.

1. Look in Windows NT's Server Control Panel for an icon labeled "GSNW" (Gateway Services for NetWare). If this icon is present, double-click on it and proceed to step 6. Otherwise, continue with step 2.

2. Double-click the Control Panel Network application and choose the Services tab.

3. In the Network application's Services tab, click on the Add button, which invokes the Select Network Services dialog box.

4. In the Select Network Services dialog box, double-click on Gateway (and Client) Services for NetWare. Windows NT will prompt you for the Windows NT Installation CD-ROM. Windows NT then will install Gateway Services for NetWare and the NWLink protocol (if it isn't already installed). If RAS is present on your system, Windows NT will ask if you want to add IPX/SPX to your RAS configuration. When the

installation is complete, shut down your system and log on again. As you log on, you might be asked to select a NetWare preferred server. Choose None and click OK.

5. Double-click on the GSNW icon in the Control Panel.

6. The Gateway Service for NetWare dialog box appears on your screen (see fig. 2.14). If you were configuring a real gateway, you would need to enter the NetWare server name in the Select Preferred Server box.

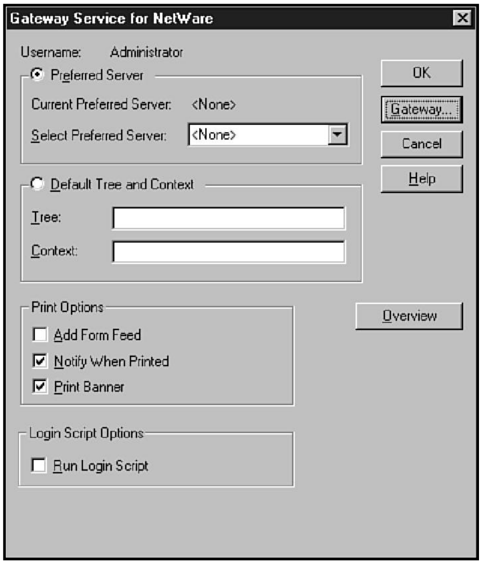


Figure 2.14

The Gateway Service for NetWare dialog box.

7. Click the Overview button for a directory of GSNW Help topics. You can browse through the topics if you have questions about configuring a NetWare gateway.

8. Click the Gateway button (refer to fig. 2.14) to configure a NetWare gateway. The Configure Gateway dialog box appears on your screen (see fig. 2.15). If you were configuring an actual gateway, you first would have to create a group called NTGATEWAY on the NetWare server, then set up a user account for the gateway on the Windows NT Server machine, and finally add the gateway user account to the NTGATEWAY group on the NetWare server. In the 9.In the

continues

Exercise 2.1: Continued

Configure Gateway dialog box, you then would check the Enable Gateway box, enter the gateway account and password, and click Add to invoke the New Share dialog box.

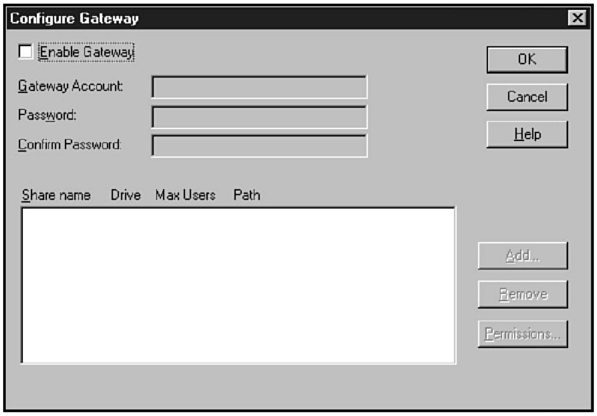


Figure 2.15

The Configure Gateway dialog box.

9. New Share dialog box, you can instruct the system to enable you to share NetWare resources on the Microsoft network. To do so, enter a share name and a network path and then enter a drive letter for a network drive mapping. In addition, you can specify a maximum number of users for the share. When you are finished, click Cancel (or click OK if you are actually connected to a NetWare network and you are setting up a real share).

10. If you configured an actual share in step 9, the share will appear in the share list of the Configure Gateway dialog box (refer to fig. 2.15). This new share will appear as a Windows NT share in the browse lists of Microsoft network clients.

note

GSNW also includes a client package, called Client Services for NetWare, which enables your computer to access NetWare resources as a NetWare client.

Exercise 2.2: SLIP and PPP in Dial-Up Networking

Objective: Explore the Dial-Up Networking application and learn how to configure Dial-Up Networking for SLIP or PPP.

Estimated time: 10 minutes

1. From the Windows NT Start menu, choose Programs/ Administrative Tools and select Dial-Up Networking. The Dial-Up Networking application enables you to connect to another computer as a dial-up client using SLIP or PPP.

note

To get the full effect of this exercise, TCP/IP, NWLink, and NetBEUI must be installed on your system and must be enabled for dial-out connections. Exercise 5.1 in Chapter 5 describes how to install protocols. To enable the protocols for dial-out connections, follow these steps:

A. Start the Control Panel Network application and select the Services tab. If Remote Access Service isn't installed, click the Add button to install it.

B. Double-click on Remote Access Service in the Services tab (or select Remote Access Service and click Properties).

C. In the Remote Access Setup dialog box, click the Configure button and make sure either the Dial Out Only or Dial Out And Receive Calls button is selected under Port Usage. Click OK.

D. In the Remote Access Setup dialog box, click the Network button. Select dial out protocols NetBEUI, TCP/IP, and IPX. Click OK.

E. Click Continue in the Remote Access Setup dialog box.

2. In the Dial-Up Networking main screen, click the New button to set up a new connection. The New Phonebook Entry dialog box appears (see fig. 2.16). The tabs of the New Phonebook Entry dialog box enable you to enter a phone

continues

Exercise 2.2: Continued

number, modem information, security information, and a login script. In addition, you can enter information about the dial-up server to which you are connecting. Click the Server tab when finished.

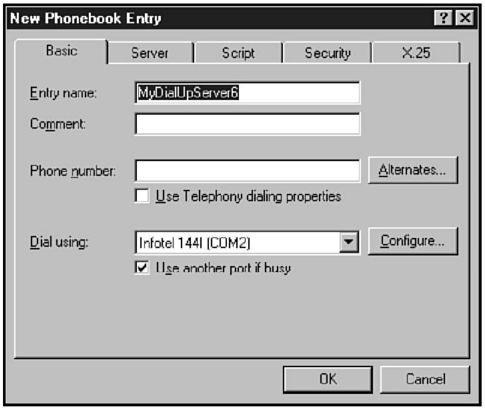


Figure 2.16

The Dial-Up Networking New Phonebook Entry dialog box enables you to enter information about the new connection.

3. In the Dial-Up Networking Server tab (see fig. 2.17), click the arrow to the right of the box labeled Dial-up server type. Note that the default option is PPP: Windows NT, Windows 95 Plus, Internet. PPP enables you to connect to a Windows NT RAS server, a Windows 95 machine with the Windows 95 Plus Dial-up Server feature, or a server with an Internet-style TCP/IP configuration.

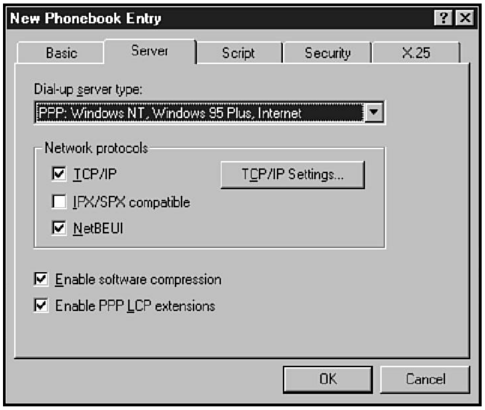


Figure 2.17

The Dial-Up Networking New Phonebook Entry Server tab.

4. In the Dial-up server type box, select the SLIP: Internet option. (TCP/IP must be installed on your machine.) Examine the rest of the Server tab options. The other protocols (IPX/SPX and NetBEUI) should be grayed out, as should software compression. Your only protocol option is TCP/IP. Check the TCP/IP check box and click the TCP/IP Settings button. Note the boxes for a static IP address and static DNS and WINS server addresses. Click Cancel.
5. In the New Phonebook Entry Server tab, click the down arrow to the right of the Dial-up server type box and choose PPP: Windows NT, Windows 95 Plus, Internet. Note that the IPX/SPX Compatible and NetBEUI protocol options are now available (if they are installed on your system and enabled for dial-out connections—see preceding note), as are software compression and PPP LCP extensions. Select the TCP/IP protocol and click the TCP/IP Settings button. Note that under a PPP connection, the TCP/IP Settings dialog box contains radio-button options for a server-assigned IP address and server-assigned name server addresses.
6. Click Cancel to exit TCP/IP Settings, Cancel to exit New Phonebook Entry, and Close to exit Dial-Up Networking.

Review Questions

The following questions test your knowledge of the information in this chapter. For additional exam help, visit Microsoft's site at www.microsoft.com/train_cert/cert/Mcpsteps.htm.

1. The OSI model organizes communication protocols into how many layers?

   A. 3

   B. 7

   C. 17

   D. 56

2. The layers of the OSI model (in order) are included in which of the following choices?

   A. Physical, Data Link, Network, Transport, System, Presentation, Application

   B. Physical, Data Link, Network, Transport, Session, Presentation, Application

   C. Physical, Data Link, Network, Transform, Session, Presentation, Application

   D. Presentation, Data Link, Network, Transport, Session, Physical, Application

3. In the OSI model, what is the relationship of a layer (N) to the layer above it (layer N+1)?

   A. Layer N provides services for layer N+1.

   B. Layer N+1 adds a header to information received from layer N.

   C. Layer N utilizes services provided by layer N+1.

   D. Layer N has no effect on layer N+1.

4. Two different computer types can communicate if __________.

   A. they conform to the OSI model
   B. they are both using TCP/IP
   C. they are using compatible protocol stacks
   D. they are a Macintosh and a Unix workstation

5. Which three of the following statements regarding protocol stacks are true?

   A. A given protocol stack can run on only one computer type.
   B. Layers add headers to packets received from higher layers in the protocol stack.
   C. A protocol stack is a hierarchical set of protocols.
   D. Each layer provides services for the next-highest layer.

6. Which protocol layer enables multiple devices to share the transmission medium?

   A. Physical
   B. MAC
   C. LLC
   D. Network

7. Which switching method employs virtual circuits?

   A. Message
   B. Circuit
   C. Packet
   D. All the above

8. Which OSI layer is concerned with data encryption?

   A. Network
   B. Transport
   C. Session
   D. Presentation

9. Which switching method makes the most efficient use of network bandwidth?

   A. Message
   B. Circuit
   C. Packet
   D. All methods are about equal

10. What is another name for a message-switching network?

    A. Connectionless
    B. Datagram
    C. Store-and-forward
    D. Virtual circuit

11. Which two statements about virtual circuits are true?

    A. They usually are associated with connection-oriented services.
    B. A virtual circuit represents a specific path through the network.
    C. A virtual circuit appears to the connected devices as a dedicated network path.
    D. Virtual circuits dedicate a communication channel to a single conversation.

12. Which three of the following terms are related?

    A. Port
    B. Connection ID
    C. Socket
    D. Service address

13. Which switching method fragments messages into small units that are routed through independent paths?

    A. Message
    B. Packet
    C. Circuit
    D. Virtual

14. Which two of the following methods of dialog control provide two-way communication?

    A. Simple duplex
    B. Simplex
    C. Half-duplex
    D. Full-duplex

15. Dialog control is a function of which layer of the OSI reference model?

    A. Network
    B. Transport
    C. Session
    D. Presentation

16. Which three of the following are functions of session administration?

    A. Connection establishment

    B. Checksum error detection

    C. Data transfer

    D. Connection release

17. Which two of the following are functions of connection establishment?

    A. Resumption of interrupted communication

    B. Verification of logon name and password

    C. Determination of required services

    D. Acknowledgment of data receipt

18. Which two of the following are possible functions of the Presentation layer?

    A. Data encryption

    B. Presentation of data on display devices

    C. Data translation

    D. Display format conversion

19. Which three of the following are possible functions of the Application layer?

    A. Network printing service

    B. End-user applications

    C. Client access to network services

    D. Service advertisement

20. PPP operates at which two of the following OSI layers?

    A. Physical

    B. Data Link

C. Network

D. Transport

21. SLIP supports which of the following transport protocols?

    A. IPX/SPX

    B. NetBEUI

    C. TCP/IP

    D. All the above

22. IEEE 802.3 is associated with which of the following network architectures?

    A. Token Ring

    B. Ethernet

    C. Internet

    D. None of the above

23. IEEE 802.5 is associated with which of the following network architectures?

    A. Token Ring

    B. Ethernet

    C. Internet

    D. None of the above

24. NDIS describes the interface between __________ and __________.

    A. user

    B. network transport protocol

    C. Physical layer

    D. network adapter driver