Chapter 5

Transport Protocols

In Chapter 2, "Networking Standards," you learned that designing network protocols usually is done in pieces, with each piece solving a small part of the overall problem. By convention, these protocols are regarded as layers of an overall set of protocols, called a *protocol suite* or a *protocol stack*.

This chapter examines a variety of actual protocols and protocol suites, such as TCP/IP, IPX/SPX, NetBEUI, AppleTalk, and DLC.

Chapter 5 targets the following objective in the Planning section of the Networking Essentials exam:

Test Objectives

- Select the appropriate network and transport protocols for various Token Ring and Ethernet networks. Protocols include the following:
  - DLC
  - AppleTalk
  - IPX
  - TCP/IP
  - NFS
  - SMB

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

1. NetBEUI operates at the _____ protocol levels.
   A. Application and Presentation
   B. Data Link and Physical
   C. Transport and Network
   D. Session and Transport

2. UDP is part of the _____ protocol suite.
   A. TCP/IP
   B. IPX/SPX
   C. AppleTalk
   D. NetBEUI

3. TCP/IP is _____ than NetBEUI.
   A. faster
   B. slower
   C. easier to install and configure
   D. none of the above

As Chapter 2 describes, the *OSI reference model* is a standard describing the activities at each level of a protocol stack. The OSI reference model is useful as a conceptual tool for understanding protocol layering. Although some protocols have been designed in strict conformance with the OSI reference model, full OSI compliance hasn't become popular. The main influence of the OSI reference model is as a conceptual framework for understanding network communication and comparing various types of protocols.

Protocols are real implementations of the conceptual rules defined in the OSI reference model. Some protocols and protocol suites existed before the OSI reference model was published and can be matched only loosely to the seven-layer model.

Packets and Protocols

Before investigating protocols and protocol stacks, take a moment to quickly review some of the protocol-related issues discussed in previous chapters.

The purpose of a network is to exchange information among computers, and protocols are the rules by which computers communicate. Computers, like humans, can adopt any number of systems for passing messages as long as the sending and receiving computers are using the same (or compatible) rules. Computers, therefore, must agree on common protocols before they can communicate—failing to do so would create a bewildering situation similar to what you'd face if you read a book in Russian to a listener who speaks only Cherokee.

The NDIS and ODI standards greatly simplify the task of finding common protocols. NDIS and ODI (described in Chapter 2) enable several protocols to operate simultaneously through the same network adapter card.

You can classify the many tasks that network protocols must oversee into a few basic categories. Think of these categories chrono-

logically, as a series of steps (each step including a collection of related tasks) that must take place before the data can reach the transmission medium. These steps are the layers of a protocol stack, as described in Chapter 2. In one sense, the term *layer* is more than metaphorical. Each layer of the stack (the Application layer, the Presentation layer, and so on) adds a layer of information to the packet, which the corresponding layer of the receiving computer needs in order to process the incoming packet.

The purpose of the layering structure is to enable vendors to adapt to specific hardware and software configurations without recreating the entire stack.

Protocols describe the way in which network data is encapsulated in packets on the source end, sent via the network to a destination, and then reconstructed at the destination into the appropriate file, instruction, or request. Breaking network data into packet-sized chunks provides smoother throughput because the small packets don't tie up the transmission medium as a larger unit of data might. Also, packets simplify the task of error detection and correction. Each file is checked separately for errors, and if an error is discovered, only that packet (instead of a whole file) must be retransmitted.

The exact composition of a network packet depends on the protocols you're using. In general, network packets contain the following:

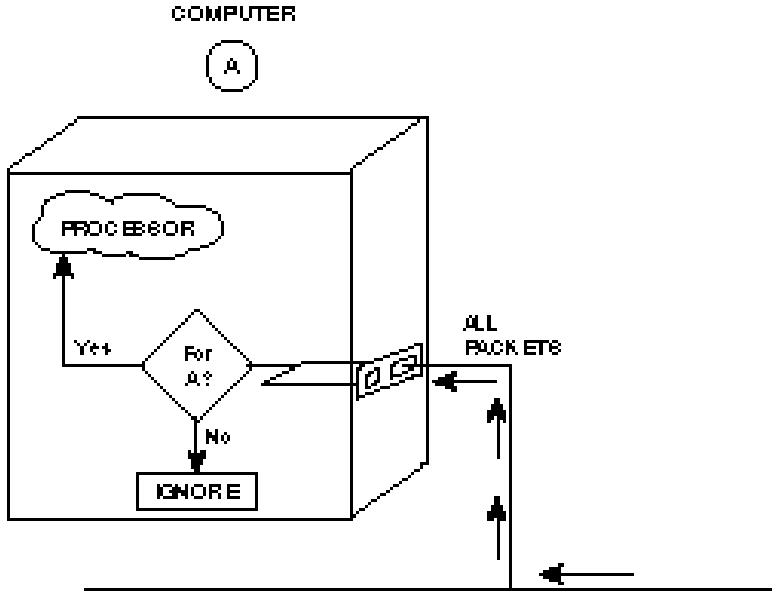
- **Header.** The header signifies the start of the packet and contains a bundle of important parameters, such as the source and destination address and time/synchronization information.
- **Data.** This portion of the packet contains the original data being transmitted.
- **Trailer.** The trailer marks the end of the packet and typically contains error-checking (Cyclical Redundancy Check, or CRC) information.

As the data passes down through the protocol layers, each layer performs its prescribed function, such as interfacing with an application, converting the data format, or adding addressing and error-checking parameters. (Chapter 2 examines the functions of the OSI protocol layers.) As you learn in this chapter, actual working protocol stacks don't always comply exactly with the OSI model—some, in fact, predate the OSI model—but the concepts and terminology of the OSI model are nevertheless useful for describing protocol functions.

When the packet reaches the transmission medium, the network adapter cards of other computers on the network segment examine the packet, checking the packet's destination address. If the destination address matches the PC's address, the network adapter interrupts the processor, and the protocol layers of the destination PC process the incoming packet (see fig. 5.1).

Figure 5.1

The network adapter card checks if the packet's destination address matches the PC's address.



Protocols and Protocol Layers

Many of the addressing, error-checking, retransmission, and acknowledgment services most commonly associated with networking take place at the Network and Transport OSI layers. (Refer to Chapter 2.) Protocol suites are often referred to by the suite's

Transport and Network protocols. In TCP/IP, for instance, TCP is a Transport layer protocol and IP is a Network layer protocol. (Note, however, that TCP/IP predates OSI and diverges from OSI in a number of ways.)

IPX/SPX is another protocol suite known by its Transport and Network layer protocols, but the order of the protocols is backward from the way the protocols are listed in TCP/IP. IPX is the Network layer protocol; SPX is the Transport layer protocol.

The lower Data Link and Physical layers provide a hardware-specific foundation, addressing items such as the network adapter driver, the media access method, and the transmission medium. Transport and Network layer protocols such as TCP/IP and IPX/SPX rest on that Physical and Data Link layer foundation, and, with the help of the NDIS and ODI standards, multiple protocol stacks can operate simultaneously through a single network adapter. (Refer to the discussion of NDIS and ODI in Chapter 2.)

Upper-level protocols provide compatibility with a particular networking environment. For instance, the so-called *NetBIOS over TCP/IP* stack provides Microsoft clients with TCP/IP.

This chapter describes the common protocol suites and many of the important protocols associated with them. In addition to TCP/IP and IPX/SPX, some of the common Transport and Network layer protocols are the following:

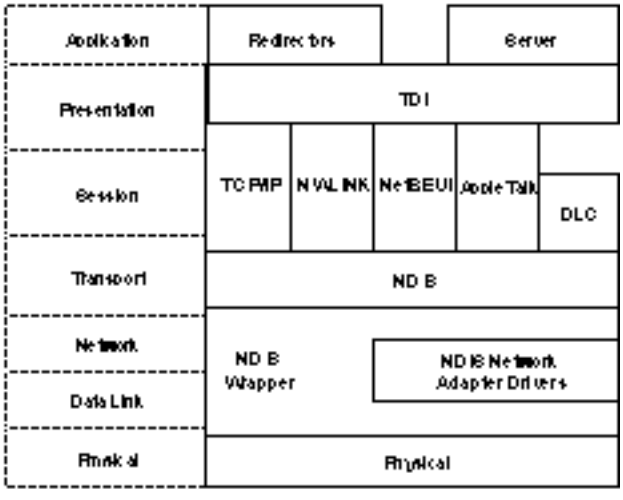
- **NWLink.** Microsoft's version of the IPX/SPX protocol essentially spans the Transport and Network layers.
- **NetBEUI.** Designed for Microsoft networks, NetBEUI includes functions at the Network and Transport layers. NetBEUI isn't routable and therefore doesn't make full use of Network layer capabilities.
- **AppleTalk Transaction Protocol (ATP) and Name Binding Protocol (NBP).** ATP and NBP are AppleTalk Transport layer protocols.
- **Datagram Delivery Protocol (DDP).** DDP is the AppleTalk Network layer protocol.

Windows NT Networking

Microsoft describes the Windows NT networking architecture as shown in figure 5.2. Note the importance of NDIS in the Windows NT networking structure. (See Chapter 2 for a description of NDIS.) The NDIS interface, NDIS wrapper, and NDIS-compatible drivers enable the TCP/IP, NWLink, NetBEUI, AppleTalk, and DLC protocols to interact simultaneously with the lower layers. (You learn more about these protocols later in this chapter.)

Figure 5.2

Windows NT networking architecture.



The Transport Driver Interface (TDI) is an interface that enables the server, redirector, and file system drivers to remain independent of the transport protocol.

NWLink is Microsoft's version of IPX/SPX.

Windows NT (like other Microsoft operating systems such as Windows for Workgroups and Windows 95) services client requests by using the Server Message Block (SMB) protocol. *SMB* is an Application layer protocol.

Three stages must take place before a protocol is operational:

1. A model describes the general function of the protocol.
2. The protocol is defined in complete detail.
3. The protocol must be realized by software and hardware designers in real products.

Consider the process of designing a building. The architect first produces sketches that describe the general nature of the building. Then the architect, possibly working with a specialist in particular building trades, develops blueprints that describe every detail of the building. Finally, an actual building is constructed.

Internet Protocols (TCP/IP)

The Internet protocol suite (also commonly called the TCP/IP protocol suite) was originally developed by the United States Department of Defense (DoD) to provide robust service on large internetworks that incorporate a variety of computer types. In recent years, the Internet protocols constitute the most popular network protocols currently in use.

One reason for the popularity of TCP/IP is that no one vendor owns it, unlike the IPX/SPX, DNA, SNA, AppleTalk protocol suites, all of which are controlled by specific companies. TCP/IP evolved in response to input from a wide variety of industry sources. Consequently, TCP/IP is the most open of the protocol suites and is supported by the widest variety of vendors. Virtually every brand of computing equipment now supports TCP/IP.

Much of the popularity of the TCP/IP protocols comes from their early availability on Unix. The protocols were built into the Berkeley Standard Distribution (BSD) Unix implementation. Since then, TCP/IP has achieved universal acceptance in the Unix community and is a standard feature on all versions of Unix.

Figure 5.3 illustrates the relationship of the protocols in the Internet suite to the layers of the OSI reference model. Notice that the suite doesn't include protocols for the Data Link or Physical layers. TCP/IP was designed to work over established standards such as Ethernet. Over time, TCP/IP has been interfaced to the majority of Data Link and Physical layer technologies.

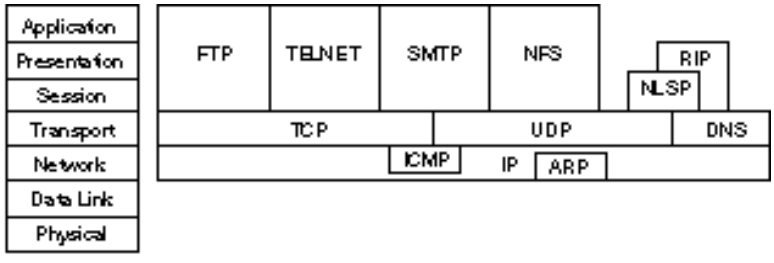


Figure 5.3

The Internet protocol suite (TCP/IP).

The Internet protocols do not map cleanly to the OSI reference model. The DoD model was, after all, developed long before the OSI model was defined. The model for the Internet protocol suite has four layers (refer to fig. 5.3). From this model, you can see the approximate relationships of the layers. The DoD model's layers function as follows:

- The Network Access layer corresponds to the bottom two layers of the OSI model. This correspondence enables the DoD protocols to coexist with existing Data Link and Physical layer standards.
- The Internet layer corresponds roughly to the OSI Network layer. Protocols at this layer move data between devices on networks.
- The Host-to-Host layer can be compared to the OSI Transport layer. Host-to-Host protocols enable peer communication between hosts on the internetwork. (At the time these protocols were designed, personal computers and workstations didn't exist, and all network computers were host computers. As a result, devices on TCP/IP networks are typically referred to as hosts. The concept of a client/server relationship didn't exist, and all communicating hosts were assumed to be peers.)

- The Process/Application layer embraces functions of the OSI Session, Presentation, and Application layers. Protocols at this layer provide network services.

One huge advantage of TCP/IP is that TCP/IP is required for communication over the Internet. One disadvantage is that the size of the protocol stack makes TCP/IP difficult to implement on some older machines. (Present-day PC models should have no problem running TCP/IP.) TCP/IP has traditionally been considered slower than other protocol stacks, but again, the power of the newer machines overcomes much of this difficulty.

A large number of protocols are associated with TCP/IP. Several of these are discussed briefly in the following sections.

Internet Protocol (IP)

The *Internet Protocol (IP)* is a connectionless protocol that provides datagram service, and IP packets are most commonly referred to as IP datagrams. IP is a packet-switching protocol that performs addressing and route selection. An IP header is appended to packets, which are transmitted as frames by lower-level protocols. IP routes packets through internetworks by utilizing dynamic routing tables that are referenced at each hop. Routing determinations are made by consulting logical and physical network device information, as provided by the Address Resolution Protocol (ARP).

IP performs packet disassembly and reassembly as required by packet size limitations defined for the Data Link and Physical layers being implemented. IP also performs error checking on the header data using a checksum, although data from upper layers is not error-checked.

Internet Control Message Protocol (ICMP)

The *Internet Control Message Protocol (ICMP)* enhances the error control provided by IP. Connectionless protocols, such as IP, cannot detect internetwork errors, such as congestion or path failures. ICMP can detect such errors and notify IP and upper-layer protocols.

Routing Information Protocol (RIP)

The *Routing Information Protocol (RIP)* in the TCP/IP suite is not the same protocol as RIP in the NetWare suite, although the two serve similar functions. Internet RIP performs route discovery by using a distance-vector method, calculating the number of hops that must be crossed to route a packet by a particular path.

Although it works well in localized networks, RIP presents many weaknesses that limit its utility on wide-area internetworks. RIP's distance-vector route discovery method, for example, requires more broadcasts and thus causes more network traffic than some other methods. The OSPF protocol, which uses the link-state route discovery method, is gradually replacing RIP. (See Chapter 6, "Connectivity Devices," for more on routing.)

Open Shortest Path First (OSPF)

The *Open Shortest Path First (OSPF)* protocol is a link-state route-discovery protocol that is designed to overcome the limitations of RIP. On large internetworks, OSPF can identify the internetwork topology and improve performance by implementing load balancing and class-of-service routing.

Transmission Control Protocol (TCP)

The *Transmission Control Protocol (TCP)* is an internetwork protocol that corresponds to the OSI Transport layer. TCP provides full-duplex, end-to-end connections. When the overhead of end-to-end communication acknowledgment isn't required, the User Datagram Protocol (UDP) can be substituted for TCP at the Transport (host-to-host) level. TCP and UDP operate at the same layer.

TCP corresponds to SPX in the NetWare environment. TCP maintains a logical connection between the sending and receiving computer systems. In this way, the integrity of the transmission is maintained. TCP detects any problems in the transmission quickly and takes action to correct them. The trade-off is that TCP isn't as fast as UDP.

TCP also provides message fragmentation and reassembly and can accept messages of any length from upper-layer protocols. TCP fragments message streams into segments that can be handled by IP. When used with IP, TCP adds connection-oriented service and performs segment synchronization, adding sequence numbers at the byte level.

In addition to message fragmentation, TCP can maintain multiple conversations with upper-layer protocols and can improve use of network bandwidth by combining multiple messages into the same segment. Each virtual-circuit connection is assigned a connection identifier called a *port*, which identifies the datagrams associated with that connection.

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is a connectionless Transport (host-to-host) layer protocol. UDP does not provide message acknowledgments; rather, it simply transports datagrams.

Like TCP, UDP utilizes port addresses to deliver datagrams. These port addresses, however, aren't associated with virtual circuits and merely identify local host processes. UDP is preferred over TCP when high performance or low network overhead is more critical than reliable delivery. Because UDP doesn't need to establish, maintain, and close connections, or control data flow, it generally outperforms TCP.

UDP is the Transport layer protocol used with the *Simple Network Management Protocol (SNMP)*, the standard network management protocol used with TCP/IP networks. UDP enables SNMP to provide network management with a minimum of network overhead.

Address Resolution Protocol (ARP)

Three types of address information are used on TCP/IP internetworks:

- **Physical addresses.** Used by the Data Link and Physical layers.

- **IP addresses.** Provide logical network and host IDs. IP addresses consist of four numbers typically expressed in dotted-decimal form. An example of an IP address is 134.135.100.13.
- **Logical node names.** Identify specific hosts with alphanumeric identifiers, which are easier for users to recall than the numeric IP addresses. An example of a logical node name is MYHOST.COM.

Given a logical node name, the Address Resolution Protocol (ARP) can determine the IP address associated with that name. ARP maintains tables of address resolution data and can broadcast packets to discover addresses on the internetwork. The IP addresses discovered by ARP can be provided to Data Link layer protocols.

Domain Name System (DNS)

The *Domain Name System (DNS)* protocol provides name and address resolution as a service to client applications. DNS servers enable humans to use logical node names to access network resources.

File Transfer Protocol (FTP)

The *File Transfer Protocol (FTP)* is a protocol for sharing files between networked hosts. FTP enables users to log on to remote hosts. Logged-on users can inspect directories, manipulate files, execute commands, and perform other commands on the host. FTP also has the capability of transferring files between dissimilar hosts by supporting a file request structure that is independent of specific operating systems.

Simple Mail Transfer Protocol (SMTP)

The *Simple Mail Transfer Protocol (SMTP)* is a protocol for routing mail through internetworks. SMTP uses the TCP and IP protocols.

SNMP doesn't provide a mail interface for the user. Creation, management, and delivery of messages to end users must be performed by an e-mail application. (The most popular e-mail application on the Internet is named Eudora.)

Remote Terminal Emulation (TELNET)

TELNET is a terminal emulation protocol. TELNET enables PCs and workstations to function as dumb terminals in sessions with hosts on internetworks. TELNET implementations are available for most end-user platforms, including Unix (of course), DOS, Windows, and Macintosh OS.

Network File System (NFS)

Network File System (NFS), developed by Sun Microsystems, is a family of file-access protocols that are a considerable advancement over FTP and TELNET. Since Sun made the NFS specifications available for public use, NFS has achieved a high level of popularity.

NFS consists of two protocols:

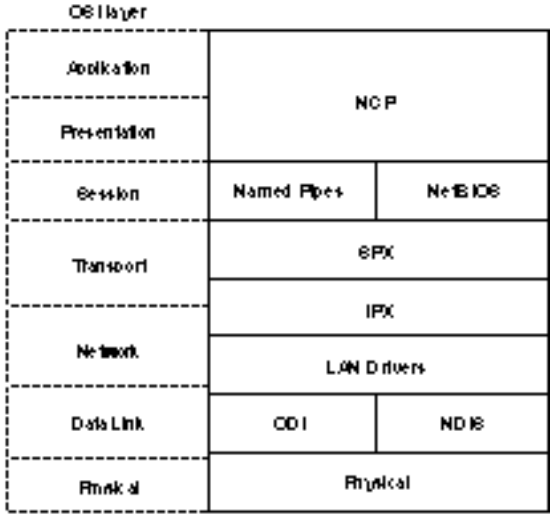
- **eXternal Data Representation (XDR).** Supports encoding of data in a machine-independent format. C programmers use XDR library routines to describe data structures that are portable between machine environments.
- **Remote Procedure Calls (RPC).** Function as a service request redirector that determines whether function calls can be satisfied locally or must be redirected to a remote host. Calls to remote hosts are packaged for network delivery and transmitted to RPC servers, which generally have the capability of servicing many remote service requests. RPC servers process the service requests and generate response packets that are returned to the service requester.

NetWare IPX/SPX

The protocols utilized with NetWare are summarized in figure 5.4. The NetWare protocols have been designed with a high degree of modularity. This modularity makes the NetWare protocols adaptable to different hardware and simplifies the task of incorporating other protocols into the suite. Windows NT doesn't use the IPX/SPX suite to communicate with NetWare resources. Microsoft instead developed a clone of IPX/SPX called NWLink IPX/SPX Compatible Transport. IPX/SPX is generally smaller and faster than TCP/IP and, like TCP/IP, it is routable.

Figure 5.4

The NetWare protocol architecture.



The *Internetwork Packet Exchange Protocol (IPX)* is a Network layer protocol that provides connectionless (datagram) service. (IPX was developed from the XNS protocol originated by Xerox.) As a Network layer protocol, IPX is responsible for internetwork routing and maintaining network logical addresses. Routing uses the RIP protocol (described later in this section) to make route selections.

IPX relies on hardware physical addresses found at lower layers to provide network device addressing. IPX also uses *sockets*, or upper-layer service addresses, to deliver packets to their ultimate destinations. On the client, IPX support is provided as a component of the older DOS shell and the current DOS NetWare requester.

The *Router Information Protocol (RIP)* uses the distance-vector route discover method to determine hop counts to other devices. Like IPX, RIP was developed from a similar protocol in the XNS protocol suite. RIP is implemented as an upper-layer service and is assigned a socket (service address). RIP is based directly on IPX and performs Network layer functions.

Sequenced Packet Exchange (SPX) is a Transport layer protocol that extends IPX to provide connection-oriented service with reliable delivery. Reliable delivery is ensured by retransmitting packets in the event of an error. SPX is derived from a similar SPX protocol in the XNS network protocol suite.

SPX establishes virtual circuits called *connections*. The connection ID for each connection appears in the SPX header. A given upper-layer process can be associated with multiple-connection IDs.

SPX is used in situations where reliable transmission of data is needed. SPX sequences the packets of data. Missing packets or packets that don't arrive in the order in which they were sent are detected immediately. In addition, SPX offers connection multiplexing, which is used in the printing environment. Many accounting programs, for example, call upon the services of SPX to ensure that data is sent accurately. On the client, SPX support is provided as a component of the older DOS shell and of the current NetWare requester.

The *NetWare Core Protocol (NCP)* provides numerous function calls that support network services, such as file service, printing, name management, file locking, and synchronization. NetWare client software interfaces with NCP to access NetWare services.

NCP is a high-level protocol built into the NetWare operating system kernel. NCP covers aspects of the Session, Presentation, and Application layers of the OSI reference model and has its own miniature language that programmers use when writing applications for the NetWare environment. The commands that NCP understands are associated primarily with access to files and directories on a file server.

NetBEUI

NetBEUI is a transport protocol that serves as an extension to Microsoft's Network Basic Input/Output System (NetBIOS). Because NetBEUI was developed for an earlier generation of DOS-based PCs, it is small, easy to implement, and fast—the fastest transport protocol available with Windows NT. Because it was built for small, isolated LANs, however, NetBEUI is non-routable, making it somewhat anachronistic in today's diverse and interconnected networking environment.

Fortunately, the NDIS standard enables NetBEUI to coexist with other routable protocols. For instance, you could use NetBEUI for fast, efficient communications on the LAN segment and use TCP/IP for transmissions that require routing (see exercise 5.2).

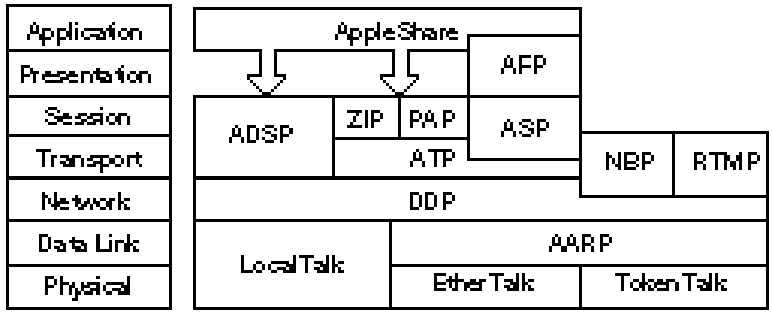
AppleTalk

AppleTalk is the computing architecture developed by Apple Computer for the Macintosh family of personal computers. Although AppleTalk originally supported only Apple's proprietary LocalTalk cabling system, the suite has been expanded to incorporate both Ethernet and Token Ring Physical layers.

AppleTalk originally supported networks of limited scope. The *AppleTalk Phase 2* specification issued in 1989, however, extended the scope of AppleTalk to enterprise networks. The Phase 2 specification also enabled AppleTalk to coexist on networks with other protocol suites. Figure 5.5 presents a layered perspective of the AppleTalk protocols.

Figure 5.5

The AppleTalk protocol suite.



The LocalTalk, EtherTalk, and TokenTalk Link Access Protocols (LLAP, ELAP, and TLAP) integrate AppleTalk upper-layer protocols with the LocalTalk, Ethernet, and Token Ring environments.

Apple's *Datagram Deliver Protocol (DDP)* is a Network layer protocol that provides connectionless service between two sockets. A *socket* is the AppleTalk term for a service address. A combination of a device address, network address, and socket uniquely identifies each process.

DDP performs network routing and consults routing tables maintained by Routing Table Maintenance Protocol (RTMP) to determine routing. Packet delivery is performed by the data link protocol operating on a given destination network.

The *AppleTalk Transaction Protocol (ATP)* is a connectionless Transport layer protocol. Reliable service is provided through a system of acknowledgments and retransmissions. Retransmissions are initiated automatically if an acknowledgment is not received within a specified time interval. ATP reliability is based on transactions. A transaction consists of a request followed by a reply. ATP is responsible for segment development and performs fragmentation and reassembly of packets that exceed the specifications for lower-layer protocols. Packets include sequence numbers that enable message reassembly and retransmission of lost packets. Only damaged or lost packets are retransmitted.

The *AppleTalk File Protocol (AFP)* provides file services and is responsible for translating local file service requests into formats required for network file services. AFP directly translates command syntax and enables applications to perform file format translations. AFP is responsible for file system security and verifies and encrypts logon names and passwords during connection setup.

AppleShare is a client/server system for Macintosh. AppleShare provides three primary application services:

- The *AppleShare File Server* uses AFP to enable users to store and access files on the network. It logs in users and associates them with network volumes and directories.
- The *AppleShare Print Server* uses NBP and PAP to support network printing. NBP provides name and address information that enables PAP to connect to printers. The AppleShare Print Server performs print spooling and manages printing on networked printers.
- The *AppleShare PC* enables PCs running MS-DOS to access AppleShare services by running an AppleShare PC program.

Data Link Control (DLC)

The Data Link Control (DLC) protocol does not provide a fully-functioning protocol stack. (Note in figure 5.2 that DLC is not continuous with the upper layers.) In Windows NT systems, DLC is used primarily to access to Hewlett Packard JetDirect network-interface printers. DLC also provides some connectivity with IBM mainframes.

The Systems Network Architecture (SNA) Protocol Suite

Another important protocol suite is IBM's Systems Network Architecture (SNA). The Microsoft BackOffice suite includes a product called SNA Server that provides connectivity with SNA networks. (The DLC protocol included with Windows NT is also sometimes used as an interface with certain SNA resources such as mainframes.)

SNA evolved when terminals were the devices usually used to interact with centralized computers. Early versions of SNA supported only hierarchical network systems designed for this centralized environment.

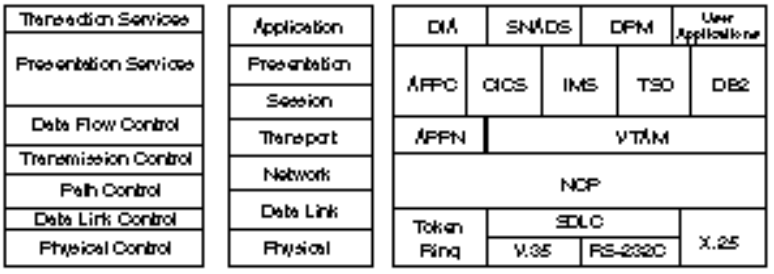
In 1984, SNA was updated to support distributed processing environments with a feature called *Advanced Peer-to-Peer Networking (APPN).* APPN can implement a distributed processing environment that can leverage the processing capabilities of mainframe hosts, minicomputers, and personal computers.

SNA wasn't developed from a preconceived, carefully thought-out model from which protocols were developed. IBM literally was pioneering the development of computer networking, and new protocols were added to meet new needs and design criteria. One result of this is that multiple protocols can be present at any given layer. Each protocol serves a somewhat different purpose in the overall scheme of SNA. As such, SNA doesn't consist of a protocol stack so much as it consists of multiple protocols that work together in different combinations to meet different needs.

SNA was a mature model by the time formulation of the OSI reference model began, and the SNA architecture had a significant influence on the definition of the OSI model. Figure 5.6 compares the layers of the OSI reference model to the layers of the SNA model.

Figure 5.6

SNA protocols and the OSI reference model.



Summary

This chapter examined network protocols and protocol suites. The chapter began with an introduction to protocol stacks. You then learned about some of the most common protocol suites, as follows:

- **TCP/IP.** The Internet protocol suite
- **IPX/SPX.** A protocol suite used for Novell NetWare networks
- **NetBEUI.** A non-routable protocol used on Microsoft networks
- **AppleTalk.** The Apple Macintosh protocol system
- **DLC.** A protocol that Windows NT networks use to connect with HP JetDirect printers and IBM mainframes

The NDIS interface standard (discussed in Chapter 2) enables a single computer to bind one network adapter to more than one protocol system. This provides great versatility and interoperability in today's diverse networking environment.

Exercises

Exercise 5.1: Installing Network Protocols in Windows NT

Objective: Become familiar with the procedure for installing and removing protocols in Windows NT.

Time estimate: 15 minutes

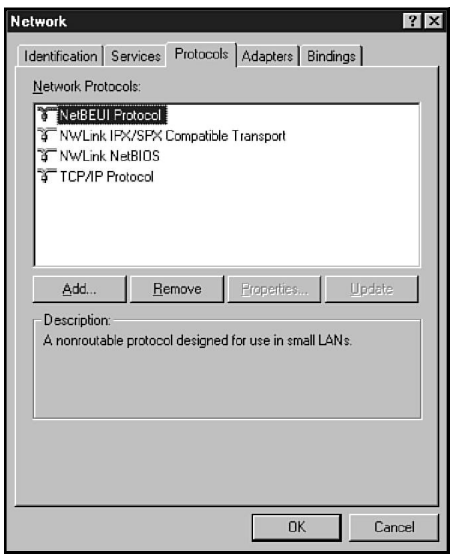
1. You can install, configure, remove, and manage network protocols by using the Network application in Windows NT's Control Panel. Click the Start menu and choose Settings/Control Panel. Then double-click on the Network application icon.

Another way to reach the Network application is to right-click on the Network Neighborhood icon and choose Properties.

2. In the Network application, choose the Protocols tab (see fig. 5.7). The Network Protocols box displays the protocols currently installed on the system.

Figure 5.7

The Network application's Protocols tab.



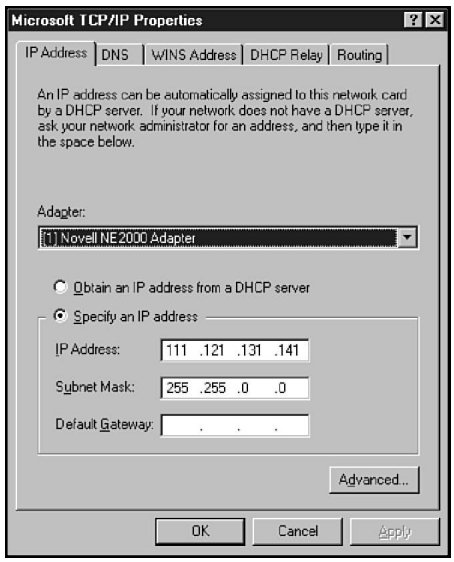
continues

Exercise 5.1: Continued

3. If TCP/IP is installed on your system, select TCP/IP Protocol and choose Properties to invoke the Microsoft TCP/IP Properties dialog box (see fig. 5.8). Note the several tabs that provide various configuration options. Close the Microsoft TCP/IP Properties dialog box and select the NetBEUI protocol (if it is installed) in the Network application's Protocols dialog box (refer to fig. 5.7). Note that the Properties button is grayed. Try double-clicking the NetBEUI icon in the box's list of protocols. A message says `Cannot configure the software component`. Unlike TCP/IP, NetBEUI is not user-configurable.

Figure 5.8

The Microsoft TCP/IP Properties dialog box.



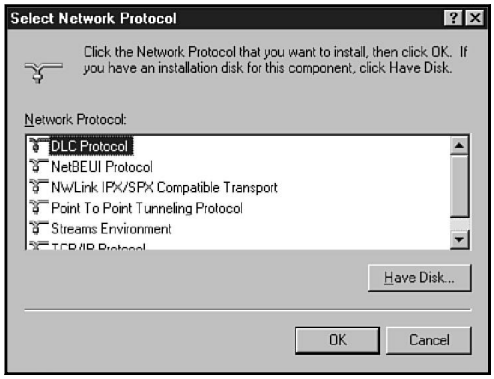
If TCP/IP and NetBEUI aren't installed on your system, you can install them by using the procedure described in steps 4 and 5 of this exercise and then delete them later.

4. To add a protocol, click on the Add button in the Network application's Protocols tab. Select a protocol from the protocol list in the Select Network Protocol dialog box (see fig. 5.9). Click on OK to install the protocol. Windows NT may prompt you for the location of the Windows NT installation

disk. If you are installing a protocol that requires some configuration (such as TCP/IP or NWLink), Windows NT will ask you for the necessary information.

Figure 5.9

The Select Network Protocol dialog box.



5. Windows NT asks you to restart your system. Shut down your system and restart. Return to the Network application's Protocols tab and see if the protocol is properly installed.

6. To remove a protocol, select the protocol from the Network Protocols list and click on the Remove button (refer to fig. 5.7).

Exercise 5.2: Network Bindings

Objective: Become familiar with the process for enabling and disabling network bindings and changing network access order.

Estimated time: 10 minutes

In Chapter 2, you learned about NDIS and the concept of network bindings. A binding is an association between protocol layers that enables those layers to behave like a protocol stack. By binding a transport protocol such as TCP/IP (which operates at the Transport and Network levels) to a network adapter (which operates at the Data Link and Physical layers) you provide a conduit for the protocol's packets to reach the network and thus enable the protocol to participate in network communications. NDIS lets you bind multiple protocols to a single adapter or multiple adapters to a single protocol.

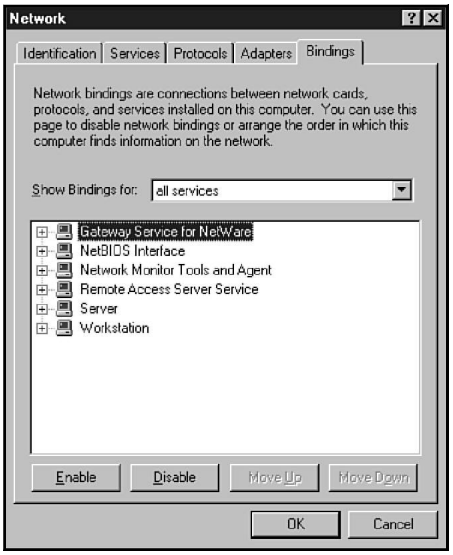
continues

Exercise 5.2: Continued

1. Click on the Start button and choose Settings/Control Panel. In Windows NT's Control Panel, double-click on the Network application icon and choose the Bindings tab (see fig. 5.10).

Figure 5.10

The Network application's Bindings tab.



2. Click on the Show Bindings for down arrow to access the drop-down list. Note that you can display bindings for services, protocols, or adapters. A service bound to a protocol bound to an adapter provides a complete pathway from the local system to the network.

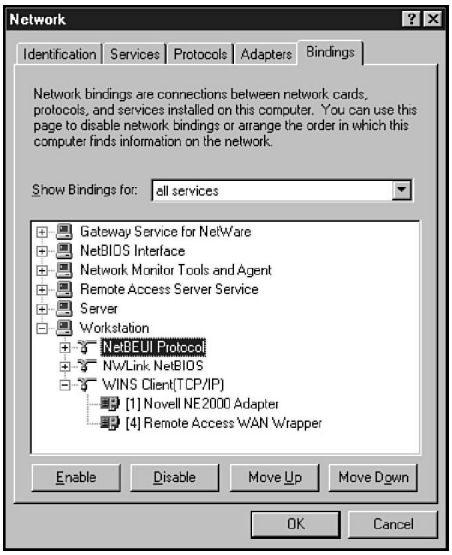
3. Click on the plus sign next to the Workstation service. The Workstation service is the Windows NT redirector (refer to Chapter 1, "Networking Terms and Concepts"), which redirects requests from the local system to the network. The protocols currently bound to the Workstation service appear in a list below the Workstation icon. Click on the plus sign next to one of the protocols. The network adapters bound to the protocol now appear in the tree (see fig 5.11).

4. The protocols and their associated adapters represent potential pathways for the Workstation service to access the network. Windows NT prioritizes those pathways according to

the order in which they appear in the Bindings tab. For the configuration shown in figure 5.11, for example, Windows NT attempts to use the NetBEUI protocol with the Workstation service before attempting to use NWLink. The Move Up and Move Down buttons let you change the access order. Select a protocol under the Workstation service. Try the Move Up and Move Down buttons to change the position of the protocol in the access order. (Don't forget to restore the protocol to its original position before leaving the Bindings tab.)

Figure 5.11

Inspect binding information by using the Bindings tab.



5. The Enable and Disable buttons let you enable or disable a protocol for a given service. Disable a protocol (for instance, NetBEUI) for the Workstation service. Now click the plus sign next to the Server service. Note that although the protocol is disabled for the Workstation service, it is still enabled for the Server service. Re-enable the protocol under the Workstation service and close the Network application.

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. Which three of the following are Transport layer protocols?

   A. ATP
   B. IPX
   C. TCP
   D. SPX

2. Which three of the following are Network layer protocols?

   A. NWLink
   B. IPX
   C. TCP
   D. IP

3. SMB operates at the _____ protocol layer.

   A. Application
   B. Transport
   C. Network
   D. Physical

4. Which three of the following protocols are available with Windows NT?

   A. AppleTalk
   B. IPX/SPX
   C. NetBEUI
   D. DLC

5. The best protocol for an isolated LAN with several DOS-based clients is _____.

   A. NWLink
   B. TCP/IP
   C. DLC
   D. NetBEUI

6. The best protocol for a remote PC that interacts with the network via the Internet is _____.

   A. NWLink
   B. TCP/IP
   C. DLC
   D. NetBEUI

7. NCP operates at the _____ protocol level(s).

   A. Application and Presentation
   B. Transport and Network
   C. Network only
   D. Transport only

8. DDP operates at the _____ protocol level(s).

   A. Application and Presentation
   B. Transport and Network
   C. Network only
   D. Transport only