

4

CHAPTER FOUR

OSI Model and Network Protocols

Objectives

1.1 Explain the function of common networking protocols

- ▶ TCP
- ▶ FTP
- ▶ UDP
- ▶ TCP/IP suite
- ▶ DHCP
- ▶ TFTP
- ▶ DNS
- ▶ HTTP(S)
- ▶ ARP
- ▶ SIP (VoIP)
- ▶ RTP (VoIP)
- ▶ SSH
- ▶ POP3
- ▶ NTP
- ▶ IMAP4
- ▶ Telnet
- ▶ SMTP
- ▶ SNMP2/3
- ▶ ICMP
- ▶ IGMP
- ▶ TLS

4.1 Explain the function of each layer of the OSI model

- ▶ Layer 1 – physical
 - ▶ Layer 2 – data link
 - ▶ Layer 3 – network
 - ▶ Layer 4 – transport
 - ▶ Layer 5 – session
 - ▶ Layer 6 – presentation
 - ▶ Layer 7 – application
-

What You Need To Know

- ▶ Identify the seven layers of the OSI model.
- ▶ Identify the function of each layer of the OSI model.
- ▶ Identify the layer at which networking devices function.
- ▶ Identify the function of various networking protocols.

Introduction

One of the most important networking concepts to understand is the *Open Systems Interconnect (OSI)* reference model. This conceptual model, created by the *International Organization for Standardization (ISO)* in 1978 and revised in 1984, describes a network architecture that allows data to be passed between computer systems.

This chapter looks at the OSI model and describes how it relates to real-world networking. It also examines how common network devices relate to the OSI model. Even though the OSI model is conceptual, an appreciation of its purpose and function can help you better understand how protocol suites and network architectures work in practical applications.

The OSI Seven-Layer Model

As shown in Figure 4.1, the OSI reference model is built, bottom to top, in the following order: physical, data link, network, transport, session, presentation, and application. The physical layer is classified as Layer 1, and the top layer of the model, the application layer, is Layer 7.

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

FIGURE 4.1 The OSI seven-layer model.

EXAM ALERT

On the Network+ exam, you might see an OSI layer referenced either by its name, such as data link, or by its layer number. For instance, you might find that a router is referred to as a Layer 3 device.

Each layer of the OSI model has a specific function. The following sections describe the function of each layer, starting with the physical layer and working up the model.

Physical Layer (Layer 1)

The physical layer of the OSI model identifies the network's physical characteristics, including the following specifications:

- ▶ **Hardware:** The type of media used on the network, such as type of cable, type of connector, and pinout format for cables.
- ▶ **Topology:** The physical layer identifies the topology to be used in the network. Common topologies include ring, mesh, star, and bus.

In addition to these characteristics, the physical layer defines the voltage used on a given medium and the frequency at which the signals that carry the data operate. These characteristics dictate the speed and bandwidth of a given medium, as well as the maximum distance over which a certain media type can be used.

Data Link Layer (Layer 2)

The data link layer is responsible for getting data to the physical layer so that it can be transmitted over the network. The data link layer is also responsible for error detection, error correction, and hardware addressing. The term *frame* is used to describe the logical grouping of data at the data link layer.

The data link layer has two distinct sublayers:

- ▶ **Media Access Control (MAC) layer:** The MAC address is defined at this layer. The MAC address is the physical or hardware address burned into each network interface card (NIC). The MAC sublayer also controls access to network media. The MAC layer specification is included in the IEEE 802.1 standard.
- ▶ **Logical Link Control (LLC) layer:** The LLC layer is responsible for the error and flow-control mechanisms of the data link layer. The LLC layer is specified in the IEEE 802.2 standard.

Network Layer (Layer 3)

The primary responsibility of the network layer is *routing*—providing mechanisms by which data can be passed from one network system to another. The network layer does not specify how the data is passed, but rather provides the mechanisms to do so. Functionality at the network layer is provided through routing protocols, which are software components.

Protocols at the network layer are also responsible for *route selection*, which refers to determining the best path for the data to take throughout the network. In contrast to the data link layer, which uses MAC addresses to communicate on the LAN, network layer protocols use software configured addresses and special routing protocols to communicate on the network. The term *packet* is used to describe the logical grouping of data at the network layer.

EXAM ALERT

the routing tables. In a dynamic routing environment, routing protocols such as *Routing Information Protocol (RIP)* and *Open Shortest Path First (OSPF)* are used. These protocols communicate routing information between networked devices on the network.

When working with networks, routes can be configured in two ways: *statically* or *dynamically*. In a static routing environment, routes are added manually to the routing

Transport Layer (Layer 4)

The basic function of the transport layer is to provide mechanisms to transport data between network devices. Primarily it does this in three ways:

- ▶ **Error checking:** Protocols at the transport layer ensure that data is sent or received correctly.
- ▶ **Service addressing:** Protocols such as TCP/IP support many network services. The transport layer makes sure that data is passed to the right service at the upper layers of the OSI model.
- ▶ **Segmentation:** To traverse the network, blocks of data need to be broken into packets that are of a manageable size for the lower layers to handle. This process, called *segmentation*, is the responsibility of the transport layer.

Protocols at the Transport Layer

Protocols that operate at the transport layer can either be connectionless, such as User Datagram Protocol (UDP), or connection-oriented, such as *Transmission Control Protocol (TCP)*. For a further discussion of these protocols, and of the difference between connection-oriented and connectionless protocols, refer to the later section “Connectionless and Connection-Oriented Protocols.”

Flow Control

The transport layer is also responsible for *data flow control*, which refers to how the receiving device can accept data transmissions. Two common methods of flow control are used:

- ▶ **Buffering:** When buffering flow control is used, data is temporarily stored and waits for the destination device to become available. Buffering can cause a problem if the sending device transmits data much faster than the receiving device can manage it.
- ▶ **Windowing:** In a windowing environment, data is sent in groups of segments that require only one acknowledgment. The size of the window (that is, how many segments fit into one acknowledgment) is defined when the session between the two devices is established. As you can imagine, the need to have only one acknowledgment for every, say, five segments can greatly reduce overhead.

Session Layer (Layer 5)

The session layer is responsible for managing and controlling the synchronization of data between applications on two devices. It does this by establishing, maintaining, and breaking sessions. Whereas the transport layer is responsible for setting up and maintaining the connection between the two nodes, the session layer performs the same function on behalf of the application.

Presentation Layer (Layer 6)

The presentation layer's basic function is to convert the data intended for or received from the application layer into another format. Such conversion is necessary because of how data is formatted so that it can be transported across the network. Applications cannot necessarily read this conversion. Some common data formats handled by the presentation layer include the following:

- ▶ **Graphics files:** JPEG, TIFF, GIF, and so on are graphics file formats that require the data to be formatted in a certain way.
- ▶ **Text and data:** The presentation layer can translate data into different formats, such as American Standard Code for Information Interchange (ASCII) and Extended Binary Coded Decimal Interchange Code (EBCDIC).
- ▶ **Sound/video:** MPEG, MP3, and MIDI files all have their own data formats to and from which data must be converted.

Another very important function of the presentation layer is *encryption*, which is the scrambling of data so that it can't be read by anyone other than the intended recipient. Given the basic role of the presentation layer—that of data-format translator—it is the obvious place for encryption and decryption to take place.

Application Layer (Layer 7)

In simple terms, the function of the application layer is to take requests and data from the users and pass them to the lower layers of the OSI model. Incoming information is passed to the application layer, which then displays the information to the users. Some of the most basic application-layer services include file and print capabilities.

The most common misconception about the application layer is that it represents applications that are used on a system such as a web browser, word processor, or spreadsheet. Instead, the application layer defines the processes that

enable applications to use network services. For example, if an application needs to open a file from a network drive, the functionality is provided by components that reside at the application layer.

EXAM ALERT

Before taking the Network+ exam, be sure you understand the OSI model and its purpose. You will almost certainly be asked questions on it for the exam.

OSI Model Summary

Table 4.1 summarizes the seven layers of the OSI model and describes some of the most significant points of each layer.

Table 4.1 OSI Model Summary

OSI Layer	Major Functions
Physical (Layer 1)	Defines the physical structure of the network and the topology.
Data link (Layer 2)	Provides error detection and correction. Uses two distinct sublayers: the Media Access Control (MAC) and Logical Link Control (LLC) layers. Identifies the method by which media are accessed. Defines hardware addressing through the MAC sublayer.
Network (Layer 3)	Handles the discovery of destination systems and addressing. Provides the mechanism by which data can be passed and routed from one network system to another.
Transport (Layer 4)	Provides connection services between the sending and receiving devices and ensures reliable data delivery. Manages flow control through buffering or windowing. Provides segmentation, error checking, and service identification.
Session (Layer 5)	Synchronizes the data exchange between applications on separate devices.
Presentation (Layer 6)	Translates data from the format used by applications into one that can be transmitted across the network. Handles encryption and decryption of data. Provides compression and decompression functionality. Formats data from the application layer into a format that can be sent over the network.
Application (Layer 7)	Provides access to the network for applications.

Identifying the OSI Layers at Which Various Network Components Operate

When you understand the OSI model, it is possible to relate the network connectivity devices discussed in Chapter 3, “Networking Components and Devices,” to the appropriate layer of the OSI model. Knowing at which OSI level a device operates allows you to better understand how it functions on the network. Table 4.2 identifies various network devices and maps them to the OSI model.

EXAM ALERT

For the Network+ exam, you are expected to be able to identify at which layer of the OSI model certain network devices operate.

Table 4.2 Mapping Network Devices to the OSI Model

Device	OSI Layer
Hub	Physical (Layer 1)
Switch	Data link (Layer 2)
Bridge	Data link (Layer 2)
Router	Network (Layer 3)
NIC	Data link (Layer 2)
Access point (AP)	Data link (Layer 2)

Connectionless and Connection-Oriented Protocols

Before getting into the characteristics of the various network protocols and protocol suites, it’s important to first identify the difference between connection-oriented and connectionless protocols.

In a *connection-oriented* communication, data delivery is guaranteed. The sending device re-sends any packet that the destination system does not receive. Communication between the sending and receiving devices continues until the transmission has been verified. Because of this, connection-oriented protocols have a higher overhead and place greater demands on bandwidth.

EXAM ALERT

Connection-oriented protocols such as TCP (Transmission Control Protocol) can accommodate lost or dropped packets by asking the sending device to retransmit them. They can do this because they wait for all the packets in a message to be received before considering the transmission complete. On the sending end, connection-oriented protocols also assume that a lack of acknowledgment is sufficient reason to retransmit.

In contrast to connection-oriented communication, *connectionless* protocols offer only a *best-effort* delivery mechanism. Basically, the information is just sent; there is no confirmation that the data has been received. If an error occurs in the transmission, there is no mechanism to re-send the data, so transmissions made with connectionless protocols are not guaranteed. Connectionless communication requires far less overhead than connection-oriented communication, so it is popular in applications such as streaming audio and video, where a small number of dropped packets might not represent a significant problem.

EXAM ALERT

As you work through the various protocols, keep an eye out for those that are connectionless and those that are connection-oriented. Also, look for protocols such as TCP that guarantee delivery of data and those such as UDP that are a fire-and-forget or best-delivery method.

Introduction to Protocols

When computers were restricted to standalone systems, there was little need for mechanisms to communicate between them. However, it wasn't long before the need to connect computers for the purpose of sharing files and printers became a necessity. Establishing communication between network devices required more than a length of cabling; a method or a set of rules was needed to establish how systems would communicate. Protocols provide that method.

It would be nice if a single protocol facilitated communication between all devices, but this is not the case. A number of protocols can be used on a network, each of which has its own features, advantages, and disadvantages. What protocol you choose can have a significant impact on the network's functioning and performance. This section explores some of the more common protocols you can expect to work with as a network administrator.

NOTE

In this chapter and throughout the book, the term request for comment (RFC) is used. RFCs are standards published by the Internet Engineering Task Force (IETF) and describe methods, behaviors, research, or innovations applicable to the operation of the Internet and Internet-connected systems. Each new RFC has an associated reference number. Looking up this number gives you information on the specific technology. For more information on RFCs, look for the Internet Engineering Task Force online.

Internet Protocol (IP)

IP, which is defined in RFC 791, is the protocol used to transport data from one node on a network to another. IP is connectionless, which means that it doesn't guarantee the delivery of data; it simply makes its best effort to do so. To ensure that transmissions sent via IP are completed, a higher-level protocol such as TCP is required.

EXAM ALERT

IP and the OSI model IP operates at the network layer of the OSI model.

In addition to providing best-effort delivery, IP also performs fragmentation and reassembly tasks for network transmissions. Fragmentation is necessary because the maximum transmission unit (MTU) size is limited in IP. In other words, network transmissions that are too big to traverse the network in a single packet have to be broken into smaller chunks and reassembled at the other end. Another function of IP is addressing. IP addressing is a complex subject. Refer to Chapter 5, “TCP/IP Routing and Addressing,” for a complete discussion of IP addressing.

Transmission Control Protocol (TCP)

TCP, which is defined in RFC 793, is a connection-oriented protocol that uses IP as its transport protocol. Being connection-oriented means that TCP establishes a mutually acknowledged session between two hosts before communication takes place. TCP provides reliability to IP communications. Specifically, TCP adds features such as flow control, sequencing, and error detection and correction. For this reason, higher-level applications that need guaranteed delivery use TCP rather than its lightweight and connectionless brother, UDP.

How TCP Works

When TCP wants to open a connection with another host, it follows this procedure:

1. It sends a message called a *SYN* to the target host.
2. The target host opens a connection for the request and sends back an acknowledgment message called an *ACK* (or *SYN ACK*).
3. The host that originated the request sends back another acknowledgment, saying that it has received the *ACK* message and that the session is ready to be used to transfer data.

When the data session is completed, a similar process is used to close the session. This three-step session establishment and acknowledgment process is called the *TCP three-way handshake*.

EXAM ALERT

TCP and the OSI model TCP operates at the transport layer of the OSI model.

TCP is a reliable protocol because it has mechanisms that can accommodate and handle errors. These mechanisms include *timeouts*, which cause the sending host to automatically retransmit data if its receipt is not acknowledged within a given time period.

User Datagram Protocol (UDP)

UDP, which is defined in RFC 768, is the brother of TCP. Like TCP, UDP uses IP as its transport protocol, but the big difference is that UDP does not guarantee delivery like TCP does. In a sense, UDP is a “fire and forget” protocol; it assumes that the data sent will reach its destination intact. In fact, the checking of whether data is delivered is left to upper-layer protocols.

EXAM ALERT

UDP and the OSI model UDP operates at the transport layer of the OSI model.

Unlike TCP, with UDP no session is established between the sending and receiving hosts, which is why UDP is called a connectionless protocol. The upshot of this is that UDP has much lower overhead than TCP. In fact, a TCP packet header has 14 fields, whereas a UDP packet header has only four fields. Therefore, UDP is much more efficient than TCP. In applications that don't need the added features of TCP, UDP is much more economical in terms of bandwidth and processing effort.

EXAM ALERT

ent protocol, and UDP is a connectionless protocol.

File Transfer Protocol (FTP)

As its name suggests, FTP provides for the uploading and downloading of files from a remote host running FTP server software. As well as uploading and downloading files, FTP allows you to view the contents of folders on an FTP server and rename and delete files and directories if you have the necessary permissions. FTP, which is defined in RFC 959, uses TCP as a transport protocol to guarantee delivery of packets.

FTP has security mechanisms used to authenticate users. However, rather than create a user account for every user, you can configure FTP server software to accept anonymous logons. When you do this, the username is anonymous, and the password normally is the user's email address. Most FTP servers that offer files to the general public operate in this way.

In addition to being popular as a mechanism for distributing files to the general public over networks such as the Internet, FTP is also popular with organizations that need to frequently exchange large files with other people or organizations. For example, the chapters in this book were sent between the author and Que Publishing using FTP. Such a system is necessary because the files we exchange are sometimes larger than can be easily accommodated using email.

EXAM ALERT

FTP and the OSI model FTP is an application layer protocol.

All the common network operating systems offer FTP server capabilities, although whether you use them depends on whether you need FTP services. All popular workstation operating systems offer FTP client functionality, although it is common to use third-party utilities such as CuteFTP and SmartFTP instead.

FTP assumes that files being uploaded or downloaded are straight text (that is, ASCII) files. If the files are not text, which is likely, the transfer mode has to be changed to binary. With sophisticated FTP clients, such as CuteFTP, the transition between transfer modes is automatic. With more basic utilities, you have to perform the mode switch manually.

Unlike some of the other protocols discussed in this chapter that perform tasks transparent to the user, FTP is an application layer service that is called upon frequently. Therefore, it can be useful to know some of the commands supported by FTP. If you are using a client such as CuteFTP, you might never need to use these commands, but they are useful to know in case you find yourself using a command-line FTP client. Table 4.3 lists some of the most commonly used FTP commands.

EXAM ALERT

FTP commands On the Network+ exam, you might be asked to identify the appropriate FTP command to use in a given situation.

Table 4.3 Commonly Used FTP Commands

Command	Description
ls	Lists the files in the current directory on the remote system.
cd	Changes the working directory on the remote host.
lcd	Changes the working directory on the local host.
put	Uploads a single file to the remote host.
get	Downloads a single file from the remote host.
mput	Uploads multiple files to the remote host.
mget	Downloads multiple files from the remote host.
binary	Switches transfers into binary mode.
ascii	Switches transfers into ASCII mode (the default).

Secure File Transfer Protocol (SFTP)

One of the big problems associated with FTP is that it is considered insecure. Even though simple authentication methods are associated with FTP, it is still susceptible to relatively simple hacking approaches. In addition, FTP transmits data between sender and receiver in an unencrypted format. By using a packet sniffer, a hacker could easily copy packets from the network and read the contents. In today's high-security computing environments, a more robust solution is needed.

That solution is the Secure File Transfer Protocol, which, based on Secure Shell (SSH) technology, provides robust authentication between sender and receiver. It also provides encryption capabilities, which means that even if packets are copied from the network, their contents remain hidden from prying eyes.

SFTP is implemented through client and server software available for all commonly used computing platforms.

EXAM ALERT

Which SFTP is it? In an industry dominated by acronyms, it should come as no surprise that eventually two protocols will have the same acronym. In this case, the SFTP acronym is used to describe both Secure File Transfer Protocol and Simple File Transfer Protocol. If you are researching additional information for the Network+ exam, make sure that you are reading about the right protocol.

Trivial File Transfer Protocol (TFTP)

A variation on FTP is TFTP, which is also a file transfer mechanism. However, TFTP does not have the security capability or the level of functionality that FTP has. TFTP, which is defined in RFC 1350, is most often associated with simple downloads, such as those associated with transferring firmware to a device such as a router and booting diskless workstations.

Another feature that TFTP does not offer is directory navigation. Whereas in FTP commands can be executed to navigate and manage the file system, TFTP offers no such capability. TFTP requires that you request not only exactly what you want but also the particular location. Unlike FTP, which uses TCP as its transport protocol to guarantee delivery, TFTP uses UDP.

EXAM ALERT

TFTP and the OSI model TFTP is an application layer protocol that uses UDP, which is a connectionless transport layer protocol. For this reason, TFTP is called a connectionless file transfer method.

Simple Mail Transfer Protocol (SMTP)

SMTP, in RFC 821, is a which is defined in RFC 821, is a protocol that defines how mail messages are sent between hosts. SMTP uses TCP connections to guarantee error-free delivery of messages. SMTP is not overly sophisticated, and it requires that the destination host always be available. For this reason, mail systems spool incoming mail so that users can read it later. How the user then reads the mail depends on how the client accesses the SMTP server.

NOTE

Sending and receiving mail SMTP can be used to both send and receive mail. Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) can be used only to receive mail.

Hypertext Transfer Protocol (HTTP)

HTTP, which is defined in RFC 2068, is the protocol that allows text, graphics, multimedia, and other material to be downloaded from an HTTP server. HTTP defines what actions can be requested by clients and how servers should answer those requests.

In a practical implementation, HTTP clients (that is, web browsers) make requests in an HTTP format to servers running HTTP server applications (that is, web servers). Files created in a special language such as Hypertext Markup Language (HTML) are returned to the client, and the connection is closed.

EXAM ALERT

HTTP and TCP HTTP is a connection-oriented protocol that uses TCP as a transport protocol. You should know this for the exam.

HTTP uses a uniform resource locator (URL) to determine what page should be downloaded from the remote server. The URL contains the type of request (for example, `http://`), the name of the server being contacted (for example, `www.microsoft.com`), and optionally the page being requested (for example, `/support`). The result is the syntax that Internet-savvy people are familiar with: `http://www.microsoft.com/support`.

Hypertext Transfer Protocol Secure (HTTPS)

One of the downsides of using HTTP is that HTTP requests are sent in clear text. For some applications, such as e-commerce, this method of exchanging information is unsuitable—a more secure method is needed. The solution is HTTPS. HTTPS uses a system known as Secure Socket Layer (SSL), which encrypts the information sent between the client and host.

For HTTPS to be used, both the client and server must support it. All popular browsers now support HTTPS, as do web server products, such as Microsoft Internet Information Server (IIS), Apache, and almost all other web server applications that provide sensitive applications. When you are accessing an application that uses HTTPS, the URL starts with `https` rather than `http`—for example, `https://www.mybankonline.com`.

Post Office Protocol Version 3/Internet Message Access Protocol Version 4 (POP3/IMAP4)

Both POP3, which is defined in RFC 1939, and IMAP4, the latest version of which is defined in RFC 1731, are mechanisms for downloading, or pulling, email from a server. They are necessary because, although the mail is transported around the network via SMTP, users cannot always read it immediately, so it must be stored in a central location. From this location, it needs to be downloaded, which is what POP and IMAP allow you to do.

POP and IMAP are popular, and many people now access email through applications such as Microsoft Outlook, Netscape Communicator, and Eudora, which are POP and IMAP clients.

One of the problems with POP is that the password used to access a mailbox is transmitted across the network in clear text. This means that if someone wanted to, he could determine your POP password with relative ease. This is an area in which IMAP offers an advantage over POP. It uses a more sophisticated

authentication system, which makes it more difficult for someone to determine a password.

EXAM ALERT

POP and IMAP POP and IMAP can be used to download, or pull, email from a server, but they cannot be used to send mail. That function is left to SMTP, which can both send and receive.

NOTE

Web-based mail: the other, other email Although accessing email by using POP and IMAP has many advantages, such systems rely on servers to hold the mail until it is downloaded to the client system. In today's world, a more sophisticated solution to any-time/anywhere email access is needed. For many people, that solution is web-based mail. Having an Internet-based email account allows you to access your mail from anywhere and from any device that supports a web browser. Recognizing the obvious advantages of such a system, all the major email systems have, for some time, included web access gateway products.

Telnet

Telnet, which is defined in RFC 854, is a virtual terminal protocol. It allows sessions to be opened on a remote host, and then commands can be executed on that remote host. For many years, Telnet was the method by which clients accessed multiuser systems such as mainframes and minicomputers. It also was the connection method of choice for UNIX systems. Today, Telnet is still commonly used to access routers and other managed network devices.

One of the problems with Telnet is that it is not secure. As a result, remote session functionality is now almost always achieved by using alternatives such as SSH.

EXAM ALERT

Telnet and UNIX/Linux Telnet is used to access UNIX and Linux systems.

Secure Shell (SSH)

Created by students at the Helsinki University of Technology, Secure Shell (SSH) is a secure alternative to Telnet. SSH provides security by encrypting data as it travels between systems. This makes it difficult for hackers using packet sniffers and other traffic-detection systems. It also provides more robust authentication systems than Telnet.

Two versions of SSH are available—SSH1 and SSH2. Of the two, SSH2 is considered more secure. Also note that the two versions are incompatible. So, if you are using an SSH client program, the server implementation of SSH that you are connecting to must be the same version.

Although SSH, like Telnet, is associated primarily with UNIX and Linux systems, implementations of SSH are available for all commonly used computing platforms, including Windows and Macintosh. As discussed earlier, SSH is the foundational technology for Secure File Transfer Protocol (SFTP).

EXAM ALERT

SSH and secure communications For the exam, you should remember that SSH is a more secure alternative to Telnet.

Internet Control Message Protocol (ICMP)

ICMP, which is defined in RFC 792, is a protocol that works with the IP layer to provide error checking and reporting functionality. In effect, ICMP is a tool that IP uses in its quest to provide best-effort delivery.

ICMP can be used for a number of functions. Its most common function is probably the widely used and incredibly useful ping utility. ping sends a stream of ICMP echo requests to a remote host. If the host can respond, it does so by sending echo reply messages back to the sending host. In that one simple process, ICMP enables the verification of the protocol suite configuration of both the sending and receiving nodes and any intermediate networking devices.

However, ICMP's functionality is not limited to the use of the ping utility. ICMP also can return error messages such as `Destination unreachable` and `Time exceeded`. (The former message is reported when a destination cannot be contacted and the latter when the time to live [TTL] of a datagram has been exceeded.)

In addition to these and other functions, ICMP performs *source quench*. In a source quench scenario, the receiving host cannot handle the influx of data at

the same rate as the data is being sent. To slow down the sending host, the receiving host sends ICMP source quench messages, telling the sender to slow down. This action prevents packets from being dropped and having to be re-sent.

ICMP is a useful protocol. Although ICMP operates largely in the background, the ping utility alone makes it one of the most valuable of the protocols discussed in this chapter.

Address Resolution Protocol (ARP)/Reverse Address Resolution Protocol (RARP)

ARP, which is defined in RFC 826, is responsible for resolving IP addresses to Media Access Control (MAC) addresses. When a system attempts to contact another host, IP first determines whether the other host is on the same network it is on by looking at the IP address. If IP determines that the destination is on the local network, it consults the ARP cache to see whether it has a corresponding entry. The ARP cache is a table on the local system that stores mappings between data link layer addresses (the MAC address or physical address) and network layer addresses (IP addresses). Here's a sample of the ARP cache:

Interface: 192.168.1.66 --- 0x8

Internet Address	Physical Address	Type
192.168.1.65	00-1c-c0-17-41-c8	dynamic
192.168.1.67	00-22-68-cb-e2-f9	dynamic
192.168.1.254	00-18-d1-95-f6-02	dynamic
224.0.0.2	01-00-5e-00-00-02	static
239.255.255.250	01-00-5e-7f-ff-fa	static

If the ARP cache doesn't have an entry for the host, a broadcast on the local network asks the host with the target IP address to send back its MAC address. The communication is sent as a broadcast because without the target system's MAC address, the source system cannot communicate directly with the target system.

Because the communication is a broadcast, every system on the network picks it up. However, only the target system replies, because it is the only device whose IP address matches the request. The target system, recognizing that the ARP request is targeted at it, replies directly to the source system. It can do this because the ARP request contains the MAC address of the system that sent it. If the destination host is determined to be on a different subnet than the sending host, the ARP process is performed against the default gateway and then repeated for each step of the journey between the sending and receiving host. Table 4.4 lists the common switches used with the arp command.

EXAM ALERT

ARP is used to link IP addressing to our Ethernet addressing (MAC addressing).

Table 4.4 Commonly Used arp Command Switches

Switch	Description
-a	Displays the entries in the ARP cache.
-s	Manually adds a permanent entry to the ARP cache.
-d	Deletes an entry from the ARP cache.

When you work with the ARP cache, you can make entries either dynamically or statically. With dynamic entries, the ARP cache is updated automatically. The ARP cache is maintained with no intervention from the user. Dynamic entries are the ones most used. Static entries are configured manually using the `arp -s` command. The static entry becomes a permanent addition to the ARP cache until it is removed using the `arp -d` command.

Reverse Address Resolution Protocol (RARP) performs the same function as ARP, but in reverse. In other words, it resolves MAC addresses to IP addresses. RARP makes it possible for applications or systems to learn their own IP address from a router or Domain Name System (DNS) server. Such a resolution is useful for tasks such as performing reverse lookups in DNS. RARP is defined in RFC 903.

NOTE

ARP functions The function of ARP is to resolve a system's IP address to the interface's MAC address on that system. Do not confuse ARP with DNS or WINS, which also perform resolution functions, but for different things.

Network Time Protocol (NTP)

NTP, which is defined in RFC 958, is the part of the TCP/IP protocol suite that facilitates the communication of time between systems. The idea is that one system configured as a time provider transmits time information to other systems that can be both time receivers and time providers for other systems.

Time synchronization is important in today's IT environment because of the distributed nature of applications. Two good examples of situations in which time synchronization is important are email and directory services systems. In

each of these cases, having time synchronized between devices is important, because without it there would be no way to keep track of changes to data and applications.

In many environments, external time sources such as radio clocks, global positioning system (GPS) devices, and Internet-based time servers are used as sources of NTP time. In others, the system's BIOS clock is used. Regardless of what source is used, the time information is communicated between devices by using NTP.

NOTE

NTP rules Specific guidelines dictate how NTP should be used. These “rules of engagement” can be found at <http://support.ntp.org/bin/view/Servers/RulesOfEngagement>.

NTP server and client software is available for a wide variety of platforms and devices. If you are looking for a way to ensure time synchronization between devices, look to NTP as a solution.

Network News Transfer Protocol (NNTP)

Network News Transfer Protocol (NNTP) is a protocol associated with posting and retrieving messages to and from newsgroups. A *newsgroup* is a discussion forum hosted on a remote system. By using NNTP client software, like that included with many common email clients, users can post, reply to, and retrieve messages.

Although web-based discussion forums are slowly replacing newsgroups, demand for newsgroup access remains high. The distinction between web-based discussion forums and NNTP newsgroups is in the fact that messages are retrieved from the server to be read. In contrast, on a web-based discussion forum, the messages are not downloaded. They are simply viewed from a remote location.

NNTP, which is defined in RFC 977, is an application layer protocol that uses TCP as its transport mechanism.

Secure Copy Protocol (SCP)

Secure Copy Protocol (SCP) is another protocol based on SSH technology. SCP provides a secure means to copy files between systems on a network. By using SSH technology, it encrypts data as it travels across the network, thereby

securing it from eavesdropping. It is intended as a more secure substitute for Remote Copy Protocol (RCP). SCP is available as a command-line utility, or as part of application software for most commonly used computing platforms.

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is a protocol that provides a mechanism to access and query directory services systems. In the context of the Network+ exam, these directory services systems are most likely to be Novell Directory Services (NDS) and Microsoft's Active Directory. Although LDAP supports command-line queries executed directly against the directory database, most LDAP interactions are via utilities such as an authentication program (network login) or locating a resource in the directory through a search utility.

Internet Group Management Protocol (IGMP)

IGMP is the protocol within the TCP/IP protocol suite that manages multicast groups. It allows, for example, one computer on the Internet to target content to a specific group of computers that will receive content from the sending system. This is in contrast to unicast messaging, in which data is sent to a single computer or network device and not to a group, or a broadcast message goes to all systems.

Multicasting is a mechanism by which groups of network devices can send and receive data between the members of the group at one time, instead of sending messages to each device in the group separately. The multicast grouping is established by each device being configured with the same multicast IP address. These multicast IP addresses are from the IPv4 Class D range, including 224.0.0.0 to 239.255.255.255 address ranges.

IGMP is used to register devices into a multicast group, as well as to discover what other devices on the network are members of the same multicast group. Common applications for multicasting include groups of routers on an internet-work and videoconferencing clients.

Domain Name System (DNS)

DNS performs an important function on TCP/IP-based networks. It resolves hostnames, such as `www.quepublishing.com`, to IP addresses, such as `209.202.161.67`. Such a resolution system makes it possible for people to remember the names of and refer to frequently used hosts using easy-to-remember hostnames rather than hard-to-remember IP addresses.

NOTE

Platform-independent Like other TCP/IP-based services, DNS is a platform-independent protocol. Therefore, it can be used on Linux, UNIX, Windows, NetWare, and almost every other platform.

In the days before the Internet, the network that was to become the Internet used a text file called `HOSTS` to perform name resolution. The `HOSTS` file was regularly updated with changes and distributed to other servers. The following is a sample of some entries from a `HOSTS` file:

```
192.168.3.45      server1  s1          #The main
                                     file and
                                     print server
192.168.3.223    mail      mailserver  #The email server
127.0.0.1        localhost
```

EXAM ALERT

The `HOSTS` file On the Network+ exam, you might be asked to identify the purpose and function of the `HOSTS` file.

NOTE

Comments in the `HOSTS` file A comment in the `HOSTS` file is preceded by a hash symbol (#).

As you can see, the host's IP address is listed, along with the corresponding host-name. It is possible to add to a `HOSTS` file aliases of the server names, which in this example are `s1` and `mailserver`. All the entries have to be added manually, and each system to perform resolutions has to have a copy of the file.

Even when the Internet was growing at a relatively slow pace, such a mechanism was both cumbersome and prone to error. It was obvious that as the network grew, a more automated and dynamic method of performing name resolution was needed. DNS became that method.

TIP

Resolution via the `HOSTS` file `HOSTS` file resolution is still supported by practically every platform. If you need to resolve just a few hosts that will not change often or at all, you can still use the `HOSTS` file for this.

DNS solves the problem of name resolution by offering resolution through servers configured to act as name servers. The name servers run DNS server software, which allows them to receive, process, and reply to requests from systems that want to resolve hostnames to IP addresses. Systems that ask DNS servers for a hostname-to-IP address mapping are called *resolvers* or *DNS clients*. Figure 4.2 shows the DNS resolution process.

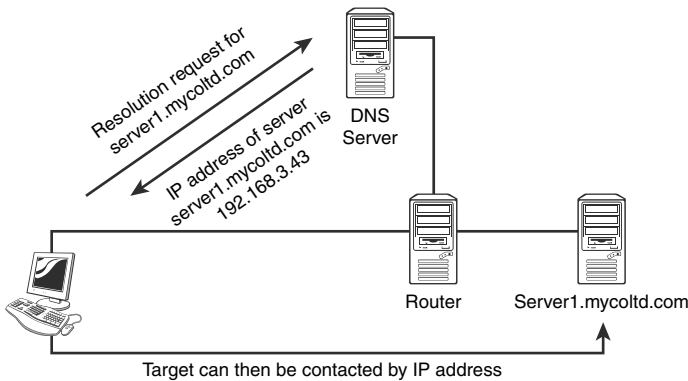


FIGURE 4.2 The DNS resolution process.

Because the DNS namespace (which is discussed in the following section) is large, a single server cannot hold all the records for the entire namespace. As a result, there is a good chance that a given DNS server might not be able to resolve the request for a certain entry. In this case, the DNS server asks another DNS server if it has an entry for the host.

NOTE

DDNS One of the problems with DNS is that, despite all its automatic resolution capabilities, entries and changes to those entries must still be performed manually. A strategy to solve this problem is to use Dynamic DNS (DDNS), a newer system that allows hosts to be dynamically registered with the DNS server.

The DNS Namespace

DNS operates in the *DNS namespace*. This space has logical divisions organized hierarchically. At the top level are domains such as .com (commercial) and .edu (education), as well as domains for countries, such as .uk (United Kingdom) and .de (Germany). Below the top level are subdomains or second-level domains associated with organizations or commercial companies, such as Red Hat and Microsoft. Within these domains, hosts or other subdomains can be assigned.

For example, the server ftp.redhat.com would be in the redhat.com domain. Or another domain called, say, development could be created, and hosts could be placed in that (that is, ftp.development.redhat.com). Figure 4.3 shows a DNS hierarchical namespace.

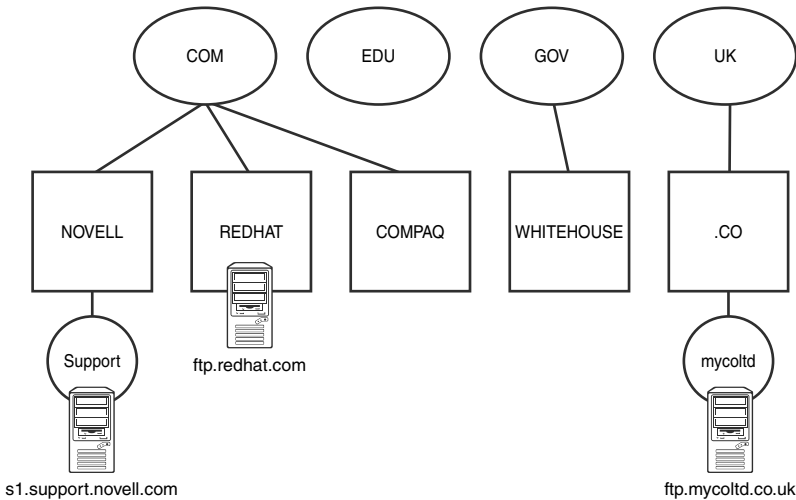


FIGURE 4.3 A DNS hierarchical namespace.

NOTE

FQDNs The domain name, along with any subdomains, is called the fully qualified domain name (FQDN) because it includes all the components from the top of the DNS namespace to the host. For this reason, many people refer to DNS as *resolving FQDNs to IP addresses*. An example of an FQDN is `www.comptia.org`, where `www` is the host, `comptia` is the second-level domain, and `.org` is the top-level domain.

The lower domains are largely open to use in whatever way the domain name holder sees fit. However, the top-level domains are relatively closely controlled. Table 4.5 lists a selection of the most widely used top-level DNS domain names. Recently, a number of top-level domains were added, mainly to accommodate the increasing need for hostnames.

Table 4.5 Selected Top-Level Domains in the DNS Namespace

Top-Level Domain Name	Intended Purpose
com	Commercial organizations
edu	Educational organizations/establishments
gov	U.S. government organizations/establishments
net	Network providers/centers
org	Not-for-profit and other organizations
mil	Military
arpa	Reverse DNS lookup
de	A country-specific domain—in this case, Germany *

* In addition to country-specific domains, many countries have created subdomains that follow roughly the same principles as the original top-level domains (such as co.uk and gov.nz).

It should be noted that although the assignment of domain names is supposed to conform to the structure shown in Table 4.5, the assignment of names is not as closely controlled as you might think. It's not uncommon for some domain names to be used for other purposes, such as .org or .net being used for business.

NOTE

Reverse lookup Although the primary function of DNS is to resolve hostnames to IP addresses, it is also possible to have DNS perform IP address-to-hostname resolution. This process is called *reverse lookup*.

Types of DNS Entries

Although the most common entry in a DNS database is an A (address) record, which maps a hostname to an IP address, DNS can hold numerous other types of entries as well. Some of particular note are the MX record, which is used to map entries that correspond to mail exchanger systems, and CNAME (canonical record name), which can be used to create alias records for a system. A system can have an A record and then multiple CNAME entries for its aliases. A DNS table with all these types of entries might look like this:

```
fileserve.mycoltd.com  IN  A    192.168.33.2
email.mycoltd.com      IN  A    192.168.33.7
fileprint.mycoltd.com  IN  CNAME fileserver.mycoltd.com
mailer.mycoltd.com     IN  MX   10  email.mycoltd.com
```

As you can see, rather than map to an actual IP address, the CNAME and MX record entries map to another host, which DNS in turn can resolve to an IP address.

DNS Records

Each DNS name server maintains information about its zone, or domain, in a series of records, known as DNS resource records. There are several DNS resource records each contain information about the DNS domain and the systems within it. These records are text entries that are stored on the DNS server. Some of the DNS resource records include:

- ▶ **Start of Authority (SOA)** The SOA is a record of information containing data on DNS zones and other DNS records. A DNS zone is the part of a domain for which an individual DNS server is responsible. Each zone contains a single SOA record.
- ▶ **Name Server (NS)** The NS record stores information that identifies the name servers in the domain that store information for that domain.
- ▶ **Canonical Name (CNAME)** The CNAME record stores additional host names, or aliases, for hosts in the domain. A CNAME specifies an alias or nickname for a canonical host name record in a domain name system (DNS) database. CNAME records are used to give a single computer multiple names (aliases).
- ▶ **Mail Exchange (MX)** The MX record stores information about where mail for the domain should be delivered.

DNS in a Practical Implementation

In a real-world scenario, whether you use DNS is almost a nonissue. If you have Internet access, you will most certainly use DNS, but you are likely to use the DNS facilities of your Internet service provider (ISP) rather than have your own internal DNS server. However, if you operate a large, complex, multiplatform network, you might find that internal DNS servers are necessary. The major network operating system vendors know that you might need DNS facilities in your organization, so they include DNS server applications with their offerings.

It is common practice for workstations to be configured with the IP addresses of two DNS servers for fault tolerance. Figure 4.4 shows an example of this.

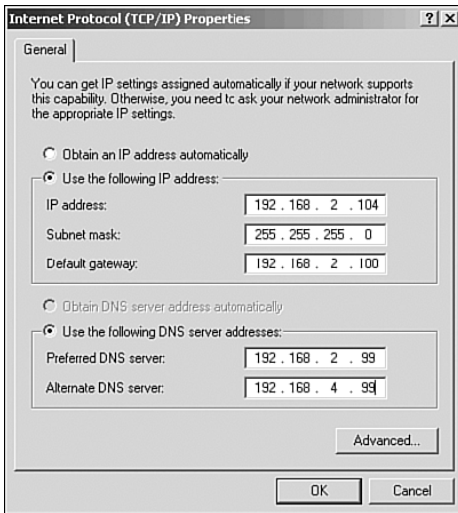


FIGURE 4.4 The DNS entries on a Windows 2003 system.

The importance of DNS, particularly in environments where the Internet is heavily used, cannot be overstated. If DNS facilities are not accessible, the Internet effectively becomes unusable, unless you can remember the IP addresses of all your favorite sites.

Windows Internet Name Service (WINS)

On Windows networks, a system called WINS enables Network Basic Input/Output System (NetBIOS) names to be resolved to IP addresses. NetBIOS name resolution is necessary on Windows networks so that systems can locate and access each other by using the NetBIOS computer name rather than the IP address. It's a lot easier for a person to remember a computer called *secretary* than to remember its IP address, 192.168.2.34. The NetBIOS name needs to be resolved to an IP address and subsequently to a MAC address (by ARP).

NetBIOS name resolution can be performed three ways on a network. The simplest way is to use a WINS server on the network that will automatically perform the NetBIOS name resolution. If a WINS server is not available, NetBIOS name resolution can be performed statically using an LMHOSTS file. Using an LMHOSTS file requires that you manually configure at least one text file with the entries. As you can imagine, this can be a time-consuming process, particularly if the systems on the network change frequently. The third method, and the default, is that systems resolve NetBIOS names using broadcasts. This approach has two problems. First, the broadcasts create additional network traffic, and second, the broadcasts cannot traverse routers unless the router is configured to forward them. This means that resolutions between network segments are impossible.

Simple Network Management Protocol (SNMP)

SNMP allows network devices to communicate information about their state to a central system. It also allows the central system to pass configuration parameters to the devices.

NOTE

SNMP is not an NMS SNMP is a protocol that facilitates network management functionality. It is not, in itself, a network management system (NMS).

Components of SNMP

In an SNMP configuration, a central system known as a *manager* acts as the central communication point for all the SNMP-enabled devices on the network. On each device that is to be managed and monitored via SNMP, software called an SNMP agent is set up and configured with the manager's IP address. Depending on the configuration, the SNMP manager then communicates with and retrieves information from the devices running the SNMP agent software. In addition, the agent can communicate the occurrence of certain events to the SNMP manager as they happen. These messages are known as *traps*. Figure 4.5 shows how an SNMP system works.

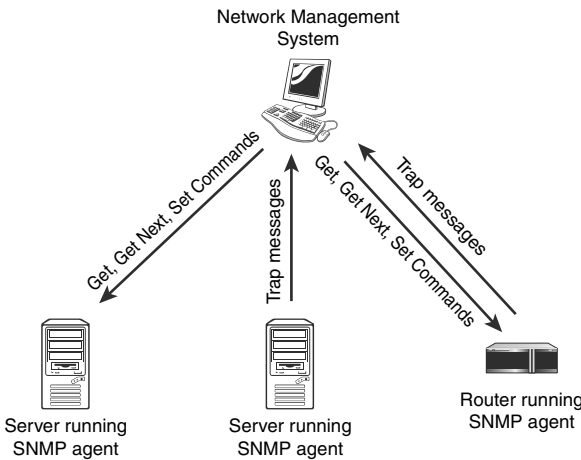


FIGURE 4.5 How SNMP works.

SNMP Management Systems

An SNMP management system is a computer running a special piece of software called a Network Management System (NMS). These software applications can be free, or they can cost thousands of dollars. The difference between the free applications and those that cost a great deal of money normally boils down to functionality and support. All NMS applications, regardless of cost, offer the same basic functionality. Today, most NMS applications use graphical maps of the network to locate a device and then query it. The queries are built into the application and are triggered by pointing and clicking. You can actually issue SNMP requests from a command-line utility, but with so many tools available, this is simply unnecessary.

NOTE

Trap managers Some people call SNMP managers or NMSs *trap managers*. This reference is misleading, however, because NMS can do more than just accept trap messages from agents.

Using SNMP and an NMS, you can monitor all the devices on a network, including switches, hubs, routers, servers, and printers, as well as any device that supports SNMP, from a single location. Using SNMP, you can see the amount of free disk space on a server in Jakarta or reset the interface on a router in Helsinki—all from the comfort of your desk in San Jose. Such power, though, brings with it some considerations. For example, because an NMS lets you reconfigure network devices, or at least get information from them, it is common practice to implement an NMS on a secure workstation platform such as a Linux or Windows server and to place the NMS PC in a secure location.

SNMP Agents

Although the SNMP manager resides on a PC, each device that is part of the SNMP structure also needs to have SNMP functionality enabled. This is performed through a software component called an *agent*.

An SNMP agent can be any device that can run a small software component that facilitates communication with an SNMP manager. SNMP agent functionality is supported by almost any device designed to be connected to a network.

As well as providing a mechanism for managers to communicate with them, agents can tell SNMP managers when a threshold is surpassed. When this happens, on a device running an SNMP agent, a trap is sent to the NMS, and the NMS then performs an action, depending on the configuration. Basic NMS sys-

tems might sound an alarm or flash a message on the screen. Other, more advanced products might send a pager message, dial a cell phone, or send an email message.

Management Information Bases (MIBs)

Although the SNMP trap system might be the most commonly used aspect of SNMP, manager-to-agent communication is not just a one-way street. In addition to being able to read information from a device using the SNMP commands `Get` and `Get Next`, SNMP managers can issue the `Set` command. Having just three commands might make SNMP seem like a limited mechanism, but this is not the case. The secret of SNMP's power is in how it uses those three commands.

To demonstrate how SNMP commands work, imagine that you and a friend each have a list on which the following four words are written: four, book, sky, and table. If you, as the manager, ask your friend for the first value, she, acting as the agent, will reply "four." This is analogous to an SNMP `Get` command. Now, if you ask for the next value, she would reply "book." This is analogous to an SNMP `Get Next` command. If you then say "set green," and your friend changes the word *book* to *green*, you will have performed the equivalent of an SNMP `Set` command. Sound simplistic? Well, if you can imagine expanding the list to include 100 values, you can see how you could navigate and set any parameter in the list, using just those three commands. The key, though, is to make sure that you and your friend have exactly the same list—which is where Management Information Bases (MIBs) come in.

SNMP uses databases of information called MIBs to define what parameters are accessible, which of the parameters are read-only, and which can be set. MIBs are available for thousands of devices and services, covering every imaginable need.

TIP

Finding a MIB Do you want to find a MIB for a device on your network? MIB Central (www.mibcentral.com) provides a searchable database of nearly 2,400 MIBs for a wide range of equipment.

To ensure that SNMP systems offer cross-platform compatibility, MIB creation is controlled by the International Organization for Standardization (ISO). An organization that wants to create a MIB can apply to the ISO. The ISO then assigns the organization an ID under which it can create MIBs as it sees fit. The assignment of numbers is structured within a conceptual model called the *hierarchical name tree*.

SNMP Communities

Another feature of SNMP that allows for manageability is communities. *SNMP communities* are logical groupings of systems. When a system is configured as part of a community, it communicates only with other devices that have the same community name. In addition, it accepts *Get*, *Get Next*, or *Set* commands only from an SNMP manager with a community name it recognizes. Typically, two communities are defined by default: a public community, intended for read-only use, and a private community, intended for read and write operations.

Whether you use SNMP depends on how many devices you have and how distributed your network infrastructure is. Even in environments that have just a few devices, SNMP can be useful, because it can act as your eyes and ears, notifying you in the event of a problem on the network.

SNMPv3

SNMP is now on its third version, and this version has some significant differences. One of the most noticeable changes is that, unlike SNMPv1 and SNMPv2, SNMPv3 supports authentication and encryption:

- ▶ **Authentication:** Authentication protocols are used to ensure that the message is from a valid source. Authentication protocols are discussed in Chapter 10, “Network Management Tools and Documentation.”
- ▶ **Encryption:** Encryption protocols are used to ensure that data cannot be read by unintended sources. Encryption protocols are discussed in Chapter 9, “Network Security.”

Dynamic Host Configuration Protocol (DHCP)

DHCP, which is defined in RFC 2131, allows ranges of IP addresses, known as *scopes*, to be defined on a system running a DHCP server application. When another system configured as a DHCP client is initialized, it asks the server for an address. If all things are as they should be, the server assigns an address from the scope to the client for a predetermined amount of time, known as the *lease*.

At various points during the lease (normally the 50% and 85% points), the client attempts to renew the lease from the server. If the server cannot perform a renewal, the lease expires at 100%, and the client stops using the address.

In addition to an IP address and the subnet mask, the DHCP server can supply many other pieces of information, although exactly what can be provided depends on the DHCP server implementation. In addition to the address information, the default gateway is often supplied, along with DNS information.

As well as having DHCP supply a random address from the scope, you can configure it to supply a specific address to a client. Such an arrangement is known as a *reservation*. Reservations are a means by which you can still use DHCP for a system but at the same time guarantee that it will always have the same IP address. DHCP can also be configured for exclusions. In this scenario, certain IP addresses are not given out to client systems.

The advantages of using DHCP are numerous. First, administrators do not have to manually configure each system. Second, human error such as the assignment of duplicate IP addresses is eliminated. Third, DHCP removes the need to reconfigure systems if they move from one subnet to another, or if you decide to make a wholesale change in the IP addressing structure. The downsides are that DHCP traffic is broadcast-based and thus generates network traffic—albeit a small amount. Finally, the DHCP server software must be installed and configured on a server, which can place additional processor load (again, minimal) on that system. From an administrative perspective, after the initial configuration, DHCP is about as maintenance-free as a service can get, with only occasional monitoring normally required.

NOTE

Platform independence DHCP is a protocol-dependent service, and it is not platform-dependent. This means that you can use, say, a Linux DHCP server for a network with Windows clients or a Novell DHCP server with Linux clients. Although the DHCP server offerings in the various network operating systems might differ slightly, the basic functionality is the same across the board. Likewise, the client configuration for DHCP servers running on a different operating system platform is the same as for DHCP servers running on the same base operating system platform.

The DHCP Process

To better understand how DHCP works, it is worth spending a few minutes looking at the processes that occur when a DHCP-enabled client connects to the network. When a system configured to use DHCP comes onto the network, it broadcasts a special packet that looks for a DHCP server. This packet is known as the DHCPDISCOVER packet. The DHCP server, which is always on the lookout for DHCPDISCOVER broadcasts, picks up the packet and compares the request with the scopes it has defined. If it finds that it has a scope for the network from which the packet originated, it chooses an address from the scope, reserves it, and sends the address, along with any other information, such as the lease duration, to the client. This is known as the DHCPOFFER packet. Because the client still does not have an IP address, this communication is also achieved via broadcast.

When the client receives the offer, it looks at the offer to determine if it is suitable. If more than one offer is received, which can happen if more than one DHCP server is configured, the offers are compared to see which is best. *Best* in this context can involve a variety of criteria but normally is the length of the lease. When the selection process is complete, the client notifies the server that the offer has been accepted, through a packet called a **DHCPREQUEST** packet. At this point the server finalizes the offer and sends the client an acknowledgment. This last message, which is sent as a broadcast, is known as a **DHCPACK** packet. After the client system receives the **DHCPACK**, it initializes the TCP/IP suite and can communicate on the network.

Transport Layer Security

The Transport Layer Security (TLS) protocol is a security protocol designed to ensure privacy between communicating client/server applications. When a server and client communicate, TLS ensures that no one can eavesdrop and intercept or otherwise tamper with the data message. TLS is the successor to SSL.

TLS is composed of two layers:

- ▶ **TLS record protocol:** Uses a reliable transport protocol such as TCP and ensures that the connection made between systems is private using data encryption.
- ▶ **TLS handshake protocol:** Used for authentication between the client and server.

NOTE

More about security Authentication, encryption, and more on security protocols are covered in Chapter 9.

Session Initiation Protocol (SIP)/Real-Time Transport Protocol (RTP)

Long-distance calls are expensive, in part because it is costly to maintain phone lines and employ technicians to keep those phones ringing. Voice over IP (VoIP) provides a cheaper alternative for phone service. VoIP technology allows regular voice conversations to occur by traveling through IP packets and via the Internet. VoIP avoids the high cost of regular phone calls by using the existing infrastructure of the Internet. No monthly bills or expensive long-distance charges are required. But how does it work?

Like every other type of network communication, VoIP requires protocols to make the magic happen. In the case of VoIP, one such protocol is Session Initiation Protocol (SIP). SIP is an application-layer protocol and is designed to establish and maintain multimedia sessions such as Internet telephony calls. This means that SIP can create communication sessions for such features as audio/videoconferencing, online gaming, and person-to-person conversations over the Internet. SIP does not operate alone; it uses TCP or UDP as a transport protocol. Remember, TCP allows guaranteed delivery of data packets, whereas UDP is a fire-and-forget transfer protocol.

EXAM ALERT

SIP operates at the application layer of the OSI model and is used to maintain a multimedia session.

TIP

SIP also includes a suite of security services, which include denial-of-service prevention, authentication (both user-to-user and proxy-to-user), integrity protection, and encryption and privacy services.

The Real-time Transport Protocol (RTP) is the Internet-standard protocol for the transport of real-time data, including audio and video. RTP can use either TCP or UDP as a transport mechanism. However, UDP is used more often, because applications using RTP are less sensitive to packet loss but typically are very sensitive to delays. UDP, then, is a faster protocol, because packet delivery is not guaranteed.

EXAM ALERT

Remember for the Network+ exam that RTP is used to transport real-time data.

RTP has two parts—the data part and the control part. The data part of RTP supports applications with real-time properties such as continuous media (such as audio and video), including timing reconstruction, loss detection, security, and content identification. The second part, RTCP, supports real-time conferencing of groups of any size within an internet.

TCP/IP Protocol Suite Summary

Table 4.6 summarizes the details of each of the protocols discussed in the preceding sections. You can use this table for review before you take the Network+ exam.

Table 4.6 TCP/IP Protocol Suite Summary

Protocol	Full Name	Description	OSI Layer
IP	Internet Protocol	A connectionless protocol used to move data around a network.	Network layer (3)
TCP	Transmission Control Protocol	A connection-oriented protocol that offers flow control, sequencing, and retransmission of dropped packets.	Transport layer (4)
UDP	User Datagram Protocol	A connectionless alternative to TCP used for applications that do not require the functions offered by TCP.	Transport layer (4)
FTP	File Transfer Protocol	Used to upload and download files to and from a remote host. Also accommodates basic file-management tasks.	Application layer (7)
SFTP	Secure File Transfer Protocol	Used to securely upload and download files to and from a remote host. Based on SSH security.	Application layer (7)
TFTP	Trivial File Transfer Protocol	A file transfer protocol that does not have the security or error checking of FTP. TFTP uses UDP as a transport protocol and therefore is connectionless.	Application layer (7)
SMTP	Simple Mail Transfer Protocol	A mechanism for transporting email across networks.	Application layer (7)
HTTP	Hypertext Transfer Protocol	Used to retrieve files from a web server.	Application layer (7)
HTTPS	Hypertext Transfer Protocol Secure	A secure protocol for retrieving files from a web server.	Application layer (7)

Table 4.6 TCP/IP Protocol Suite Summary *continued*

Protocol	Full Name	Description	OSI Layer
POPv3/ IMAPv4	Post Office Protocol version 3/Internet Message Access Protocol version 4	Used to retrieve email from a server on which it is stored. Can only be used to retrieve mail. IMAP and POP cannot be used to send mail.	Application layer (7)
Telnet	Telnet	Allows sessions to be opened on a remote host.	Application layer (7)
SSH	Secure Shell	Allows secure sessions to be opened on a remote host.	Application layer (7)
ICMP	Internet Control Message Protocol	Used on IP-based networks for error reporting, flow control, and route testing.	Network layer (3)
ARP	Address Resolution Protocol	Resolves IP addresses to MAC addresses to enable communication between devices.	Data link layer (2)
RARP	Reverse Address Resolution Protocol	Resolves MAC addresses to IP addresses.	Data link layer (2)
NTP	Network Time Protocol	Used to communicate time synchronization information between devices.	Application layer (7)
NNTP	Network News Transport Protocol	Facilitates the access and downloading of messages from newsgroup servers.	Application layer (7)
SCP	Secure Copy Protocol	Allows files to be copied securely between two systems. Uses SSH technology to provide encryption services.	Application layer (7)
LDAP	Lightweight Directory Access Protocol	Used to access and query directory services systems such as Novell Directory Services and Microsoft Active Directory.	Application layer (7)

Table 4.6 TCP/IP Protocol Suite Summary *continued*

Protocol	Full Name	Description	OSI Layer
IGMP	Internet Group Management Protocol	Provides a mechanism for systems within the same multicast group to register and communicate with each other.	Network layer (3)
DNS	Domain Name System	Resolves hostnames to IP addresses.	Application layer (7)
DHCP	Dynamic Host Configuration Protocol	Automatically assigns TCP/IP information.	Application layer (7)
SNMP	Simple Network Management Protocol	Allows network devices to communicate information about their state to a central system. It also allows the central system to pass configuration parameters to the devices.	Application layer (7)
TLS	Transport Layer Security	A security protocol designed to ensure privacy between communicating client/server applications.	Application layer (7)
SIP	Session Initiation Protocol	An application layer protocol designed to establish and maintain multimedia sessions such as Internet telephony calls.	Application layer (7)
RTP	Real-time Transport Protocol	The Internet-standard protocol for transporting real-time data.	Application layer (7)

Review and Test Yourself

The following sections provide you with the opportunity to review what you've learned in this chapter and to test yourself.

The Facts

- ▶ The application layer provides access to the network for applications and certain user functions. It displays incoming information and prepares outgoing information for network access.
- ▶ The presentation layer converts data from the application layer into a format that can be sent over the network. It converts data from the session layer into a format that the application layer can understand. It encrypts and decrypts data. It provides compression and decompression functionality.
- ▶ The session layer synchronizes the data exchange between applications on separate devices. It handles error detection and notification to the peer layer on the other device.
- ▶ The transport layer establishes, maintains, and breaks connections between two devices. It determines the ordering and priorities of data. It performs error checking and verification and handles retransmissions if necessary.
- ▶ The network layer provides mechanisms for routing data between devices across single or multiple network segments. Handles the discovery of destination systems and addressing.
- ▶ The data link layer has two distinct sublayers: LLC and MAC. The LLC performs error detection and handling for the transmitted signals and defines the method by which the medium is accessed. It also defines hardware addressing through the MAC sublayer.
- ▶ The physical layer defines the network's physical structure. It defines voltage/signal rates, the physical connection methods, and the physical topology.
- ▶ Application protocols map to the application, presentation, and session layers of the OSI model. These include FTP, TFTP, SMTP, HTTP, and SNMP.
- ▶ Transport protocols map to the transport layer of the OSI model and are responsible for transporting data across the network. UDP is a transport protocol.

Key Terms

- ▶ OSI
- ▶ Physical layer
- ▶ Data link layer
- ▶ Network layer
- ▶ Transport layer
- ▶ Session layer
- ▶ Presentation layer
- ▶ Application layer
- ▶ LLC
- ▶ MAC
- ▶ Static routing
- ▶ Dynamic routing
- ▶ TCP
- ▶ UDP
- ▶ Connectionless protocol
- ▶ Connection-oriented protocol
- ▶ Protocol suite
- ▶ Application protocol
- ▶ Transport protocol
- ▶ Network protocol
- ▶ Packet
- ▶ FTP
- ▶ SNMP
- ▶ SMTP
- ▶ TCP
- ▶ UDP
- ▶ IP
- ▶ TCP/IP addressing