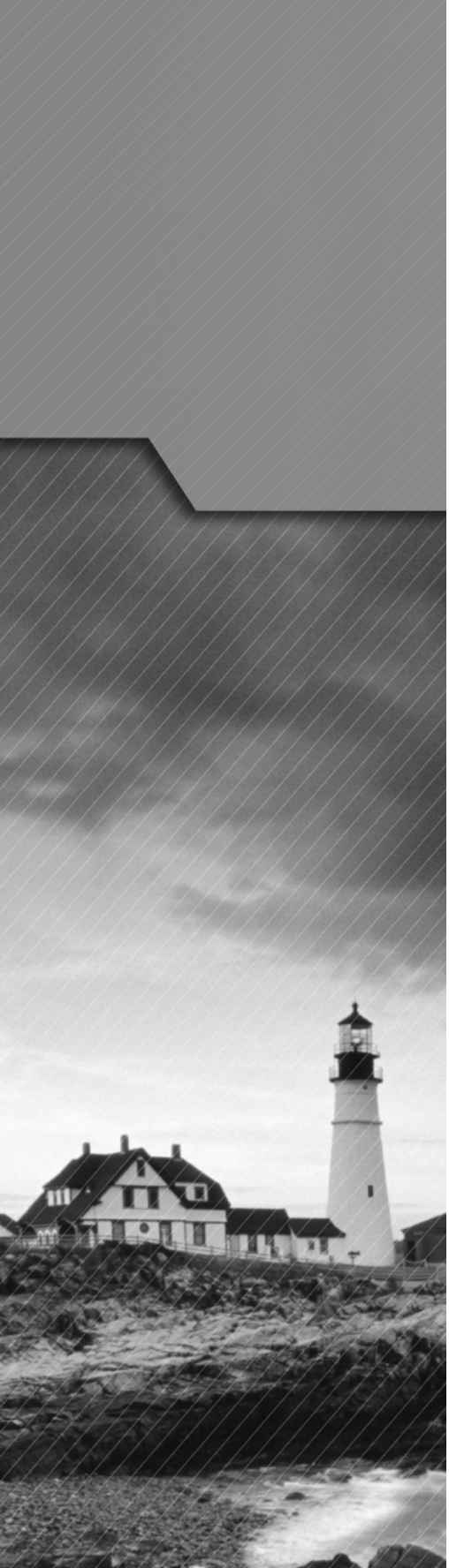
Chapter

1

Domain 1 Network Technologies

COMPTIA NETWORK+ EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ **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
  - SMNPv2/3
  - ICMP
  - IGMP
  - TLS



✓ **1.2 Identify commonly used TCP and UDP default ports**

- TCP ports
- FTP – 20, 21
- SSH – 22
- TELNET – 23
- SMTP – 25
- DNS – 53
- HTTP – 80
- POP3 – 110
- NTP – 123
- IMAP4 – 143
- HTTPS – 443
- UDP ports
- TFTP – 69
- DNS – 53
- BOOTPS/DHCP – 67
- SNMP – 161

✓ **1.3 Identify the following address formats**

- IPv6
- IPv4
- MAC addressing

✓ **1.4 Given a scenario, evaluate the proper use of the following addressing technologies and addressing schemes**

- Subnetting
- Classful vs. classless (e.g., CIDR, Supernetting)
- NAT
- PAT
- SNAT
- Public vs. Private
- DHCP (static, dynamic, APIPA)
- Unicast
- Multicast
- Broadcast

- ✓ **1.5 Identify common IPv4 and IPv6 routing protocols**
  - Link state
  - OSPF
  - IS-IS
  - Distance vector
  - RIP
  - RIPv2
  - BGP
  - Hybrid
  - EIGRP
- ✓ **1.6 Explain the purpose and properties of routing**
  - IGP vs. EGP
  - Static vs. dynamic
  - Next hop
  - Understanding routing tables and how they pertain to path selection
  - Explain convergence (steady state)
- ✓ **1.7 Compare the characteristics of wireless communication standards**
  - 802.11 a/b/g/n
  - Speeds
  - Distance
  - Channels
  - Frequency
  - Authentication and encryption
  - WPA
  - WEP
  - RADIUS
  - TKIP

In every network, three components are essential in order for computers to be able to communicate: a common protocol, a common network media, and a common network client or service. In this chapter, I'll discuss the first component — the protocol. Although many types of protocols are in use today, all protocols have one element in common: they are a set of rules by which a network or a group of components behave in order to communicate.

The types of protocols you utilize will depend largely on the type of network you are using. Some protocols are much more common than others. Many protocols can stand on their own, whereas other protocols are part of a larger suite of protocols. You can use protocols to facilitate as well as to secure communication, but ultimately you must understand protocols in order to make effective use of them. You should be aware of the many different protocols in use today and understand how they work together and, in some cases, how they don't work together.

In this chapter, I'll start by discussing the factors that protocols have in common and how you can identify different types of protocols. I will also identify the types of network components that are most likely to use each type of protocol. After I have discussed the commonalities of protocols, you will then turn your attention to the differences in various protocols. Later, I will also define each of the protocols as it relates to the entire model of communication, namely, the Open Systems Interconnect (OSI) model. You should understand protocols in general terms as well as the many specific protocols in various protocol suites.

For more detailed information on these topics, please see *Network+ Study Guide*, published by Wiley.

1.1 Explain the function of common networking protocols

As I've discussed, protocols are sets of rules that determine how communication will take place. In regard to networks, you might think of them as a language that computers use to "talk" to one another. If two devices speak the same language, then they can understand each other. In addition, groups of protocols are combined to create protocol suites. One of the most important protocol suites in today's networks is Transmission Control Protocol/Internet Protocol (TCP/IP).

The TCP/IP protocol suite contains many protocols. These protocols work together to provide communication, management, diagnostics, and troubleshooting for a network that uses the TCP/IP protocol. To understand TCP/IP, it is essential that you understand all the protocols in the suite.

In the following sections, I will define the purpose, function, and use of each of the protocols in the TCP/IP protocol suite. In addition, I will discuss the TCP/IP protocol layers and define the layer at which each of the protocols operates. I will also discuss how the TCP/IP protocol loosely aligns with the OSI model of communication. Table 1.1 summarizes the essential elements of each of these protocols, which are covered next.

TABLE 1.1 Characteristics of Protocols in the TCP/IP Protocol Suite

Protocol	Purpose	Function	Use
IP	Addresses and transports data from one network node to another.	A Network layer connectionless protocol, it "fires and forgets." Performs fragmenting and assembling of packets.	IP addresses are assigned to computers and to router interfaces. These addresses are used to transfer a packet into the proper network so it can be delivered to a host.
TCP	Responsible for flow control and error recovery.	Waits for receipt of acknowledgments from the destination that packets have been delivered without errors. Resends packets that are not acknowledged within a specified time frame. Works at the Transport layer of the TCP/IP suite.	Used with protocols that require a guaranteed delivery such as FTP, HTTP, SMTP, and others.
UDP	Broadcasts packets through a network making a "best effort" to deliver them to the destination.	Connectionless protocol. Works at the Transport layer of the TCP/IP suite.	Used for applications that can provide their own acknowledgments or can be monitored, such as multimedia over the internet.
FTP	Provides the rules of behavior for transferring files through an intranet or over the Internet.	Works at the Application layer of the TCP/IP suite. Provides a protocol as well as an application for transferring files.	Used to browse file structures on a remote computer and to transfer files between computers within intranets and on the Internet.

TABLE 1.1 Characteristics of Protocols in the TCP/IP Protocol Suite *(continued)*

Protocol	Purpose	Function	Use
TFTP	Provides for transferring files within a network.	Connectionless protocol that works at the Application layer. Uses UDP for low overhead without a guarantee of delivery.	Typically used for simple file transfers such as those between a computer and a router or a switch for management purposes.
SMTP	Provides for the delivery of mail messages within a network or between networks.	Works at the Application layer and uses TCP to guarantee delivery of mail to remote hosts.	Typically used to transfer email messages within a network and between networks.
HTTP	Provides for browsing services for the World Wide Web.	Works at the Application layer and provides access to files on web servers through the use of URLs to pages that are formatted web languages such as HTML.	Typically used to browse information on the many servers that interconnect the World Wide Web.
HTTPS	Provides for access to resources on the Internet in a secure fashion.	Works at the Application layer and uses SSL to encrypt data traffic so communications on the Internet can remain secure.	Used for Internet communications that must remain secure, such as banking, e-commerce, and medical transactions.
POP3	Allows the storage and retrieval of user email on servers. Allows users to access and download email from servers.	Works at the Application layer. Users can connect to the server and download messages to a client. The messages can then be read of the client.	Used for many email applications. User can check their email boxes and download messages that have been placed in them.
IMAPv4	Allows the storage and retrieval of user email on servers. Allows users to access email on servers and either read the email on the server or download the email to the client to read it.	Works at the Application layer of the TCP/IP suite. Allows a user to read messages on an email server without the need to download the messages off the server.	Typically, this method of email retrieval is convenient for users who travel and therefore might access their email from more than one location. The mail remains on the server until they delete it, so they can gain access to it from multiple locations.

TABLE 1.1 Characteristics of Protocols in the TCP/IP Protocol Suite *(continued)*

Protocol	Purpose	Function	Use
Telnet	Provides a virtual terminal protocol for connecting to a managing server.	Works at the Application layer of the TCP/IP suite. Provides a connection using an authentication method that is performed in clear text. This protocol and application are not considered secure.	Has been used in the past for "dumb terminals" that connected to mainframe computers. Is now used to connect computers to servers, routers, switches, and so on, for remote management.
SSH	Provides the capability to log onto a computer remotely, execute commands, and move files in a secure and encrypted environment.	Works at the Application layer of the TCP/IP suite. Provides for a secure logon and a secure environment in which to execute commands.	Typically used to manage servers from clients and to move sensitive files from one server to another within the same network or between networks.
ICMP	Provides error checking and reporting functionality.	Works at the Internet layer of the TCP/IP suite. Provides background services that can be used to provide information to an administrator and to request a "quench" of the information flow in the network.	Typically used as part of the ping tool to test network connectivity. Can send back an echo reply when an echo request message is sent to it. Can also send back a message such as "Destination Host Unreachable" and "Time Exceeded" when the connection to the "pinged" host is not possible.
ARP	Resolves IP addresses to MAC addresses.	Works at the Internet layer of the TCP/IP suite. Includes a cache that is checked first. If the entry is not found in the cache, then ARP uses a broadcast to determine the MAC address of the client.	Typically used by the system as a background service but also includes a utility that can be used for troubleshooting.
RARP	Resolves IP addresses to MAC addresses.	Works at the Internet layer of the TCP/IP suite. It assigns an IP address when presented with a MAC address.	Used with diskless workstations to assign an IP address automatically. Also sometimes used as very rudimentary security for computer authentication.

TABLE 1.1 Characteristics of Protocols in the TCP/IP Protocol Suite *(continued)*

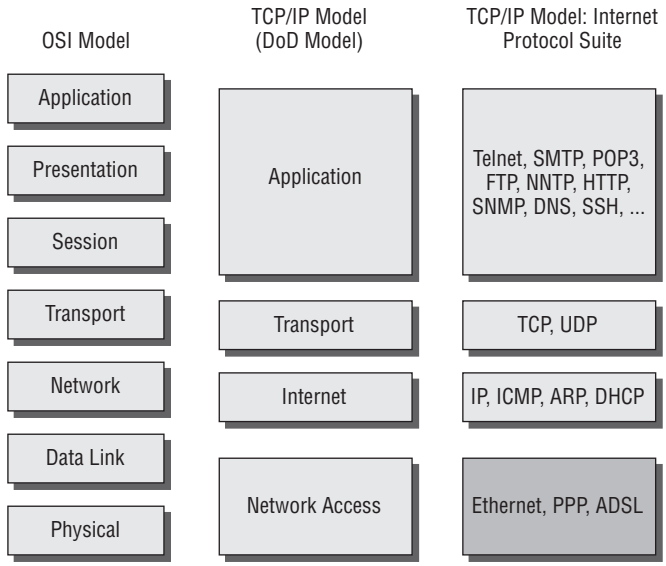
Protocol	Purpose	Function	Use
NTP	Synchronizes time between computers in a network.	Works at the Application layer of TCP/IP suite. Can synchronize time between clients and servers.	Used to synchronize time to assure that authentication protocols such as the Kerberos protocol work properly and that applications that require collaboration operate properly.
NNTP	Provides access to the USENET newsgroups on news servers.	Works at the Application layer of the TCP/IP suite. Provides a set of standards for accessing and opening news articles on a USENET-based news server.	Typically used by individuals and organizations to research information about a variety of topics. News servers do not provide for "browsing" but instead provide lists of articles for specified topics.
SIP	Sets up and tears down voice and video calls over the Internet.	Works at the Session layer of the OSI model and the Application layer of the TCP/IP suite.	Typically used for Voice over IP (VoIP) and video communications.
RTP	Defines a standardized packet format for delivering audio and video over the Internet.	Works at the Session layer of the OSI model and the Application layer of the TCP/IP suite.	Used to enhance multimedia communications for streaming, video conferencing, and push-to-talk applications.
IGMP	Provides a standard for multicasting on an intranet.	Allows a host to inform its local router, using Host Membership Reports that it wants to receive messages addressed to a specific multicast group.	Used to establish host memberships in multicast groups on a single network.
TLS	A network security protocol that provides for data confidentiality and integrity.	Works through active peer negotiation of authentication and encryption protocols.	Used for secure transmission of data between servers and clients within a network and between networks.

Note: The OSI layer names and numbers will be covered in more detail in Chapter 4, "Network Management."

It's important to understand that TCP/IP is not just one protocol, or even two protocols, but it is instead an entire group of protocols that work together to support network communication. Although the OSI model is just a model, the TCP/IP suite represents the

continual development of protocols, each of which loosely aligns itself to a portion of the OSI model. The Department of Defense (DOD) defines the TCP/IP suite of protocols as having four layers, as shown in Figure 1.1. Each of the protocols in the TCP/IP suite can be said to function in one or more of these layers. In the following sections, I'll discuss the most common of these protocols.

FIGURE 1.1 The TCP/IP protocol suite



Internet Protocol (IP)

IP is a protocol that is used to transport data from one node on a network to another node. A node can be a computer or a router interface. IP is considered to be a *connectionless* protocol, which works at the Network layer of the OSI model. Because it is connectionless, it does not establish a session with another computer and does not guarantee the delivery of packets; it only makes an effort to deliver them. To guarantee the delivery of packets, a higher-level protocol such as TCP is required.

IP also performs the task of fragmenting and reassembling packets when needed. *Fragmentation* is sometimes necessary because devices that make up the network have a maximum transmission unit (MTU) size that is smaller than the packet to be delivered. In this case, the packet must be "broken up" into smaller pieces and then reassembled on the other side of the transmission. This is an important role that IP provides for the network.

Probably the most widely known role that IP provides is the addressing of packets. IP marks each packet with a source address and a destination address. IP addressing is essential to the success of network communications. For example, when you see a number assigned to a computer's location in a network, such as 192.168.0.1, you are looking at an IP address for this device. I will discuss more IP addressing functions later in this chapter.

Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol that works at the Transport layer (layer 4) of the OSI model. It uses IP as its transport protocol and assists IP by providing a guaranteed mechanism for delivery. TCP requires that a session first be established between two computers before communication can take place. TCP also adds features such as flow control, sequencing, and error detection and correction. This guaranteed delivery mechanism is a requirement in order for TCP to operate at all. For this reason, you should understand how TCP operates.

TCP works by a process referred to as a *three-way handshake*. The TCP three-way handshake works as follows:

1. TCP sends a short message called a SYN to the target host.
2. The target hosts opens a connection for the request and sends back an acknowledgment message called a SYN ACK.
3. The host that originated the request sends back another acknowledgment called an ACK, confirming that it has received the SYN ACK message and that the session is ready to be used to transfer data.

A similar process is used to close the session when the data exchange is complete. The entire process provides a reliable protocol. TCP extends its reliability by making sure that every packet it sends is acknowledged. If a packet is not acknowledged within the timeout period, the packet is resent automatically by TCP. The only disadvantage of a connection-oriented protocol is that the overhead associated with the acknowledgments tends to slow it down.

User Datagram Protocol (UDP)

UDP also operates at the Transport layer of the OSI model and uses IP as its transport protocol, but it does not guarantee the delivery of packets. It doesn't guarantee the delivery of packets because UDP does not establish a session. UDP is instead known as a "fire and forget" protocol because it assumes that the data sent will reach its destination and does not require acknowledgments. Because of this, UDP is also referred to as a *connectionless protocol*.

Now you might be wondering why anyone would want to use UDP instead of TCP. The advantage of UDP is its low overhead in regard to bandwidth and processing effort. Whereas a TCP header has 11 fields of information that have to be processed, a UDP header has only 4 fields. Applications that can handle their own acknowledgments and that do not require the additional features of the TCP protocol might use the UDP protocol to take advantage of the lower overhead. Multimedia presentations that are broadcast or multicast onto the network often use UDP since they can be monitored to make sure that the packets are being received. Services such as the Domain Name System (DNS) service also take advantage of the lower overhead provided by UDP.

Dynamic Host Configuration Protocol (DHCP)

DHCP is actually more of a service than a protocol. When a client comes on to a network, it needs an IP address. You could statically assign every computer in your network, but that would be doing it the hard way. The easier and smarter way would be to use the DHCP protocol (service) to make automatic assignments for you. You can even configure a DHCP server to give a client other information, such as the address of the DNS server.

All Microsoft clients since Windows 98 have their default installation configurations set to obtain an IP address automatically. They are already looking for a DHCP server when they start up. When you include a properly configured DHCP server on your network, you avoid a great number of IP misconfigurations and save yourself a lot of manual labor.

Domain Name System (DNS)

DNS is a service and a protocol. It uses relational databases to resolve hostnames of computers and other network clients to their assigned IP addresses. DNS facilitates "friendly naming" of resources on a network and on the Internet so you don't have to remember, for example, the IP address for MSNBC.com. Clients can be statically configured with the addresses of the DNS servers that host the DNS database, or the DHCP server can provide that information to the client.

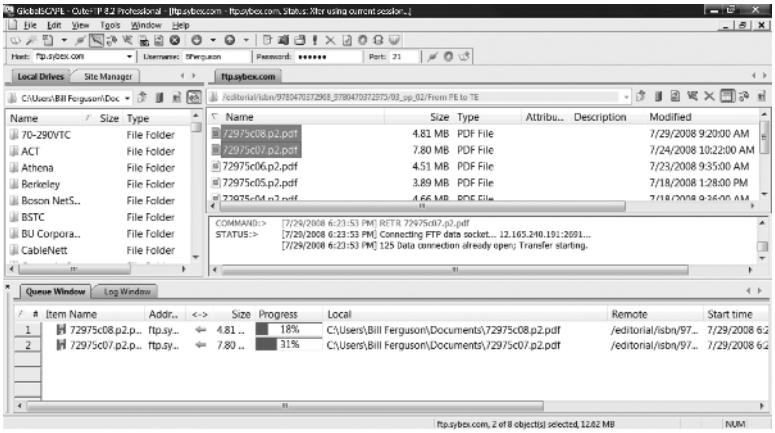
File Transfer Protocol (FTP)

FTP, as its name indicates, provides for the transfer of files through a network environment. It can be used within an intranet or through the Internet. FTP is actually more than just a protocol; it is an application as well, and thus FTP works at the Application layer of the OSI model and uses the TCP protocol as a transport mechanism. FTP allows a user to browse a folder structure on another computer (assuming they have been given the permissions to authenticate to the computer) and then to download files from the folders or to upload additional files.

Many organizations use FTP to make files available to the general public and therefore allow users to log on to the FTP server anonymously. In other words, the users do not have to utilize a username and password to authenticate to the server. Since the files are there for the public, the users are allowed to access them without authenticating. Organizations also use FTP to transfer files within an organization. Typically, these servers require authentication by the user, either by supplying an additional username and password or by using a pass-through authentication provided by a previous logon, such as to Active Directory.

You can use FTP through most browsers and even from a command line, but it is quite common for users to purchase third-party software such CuteFTP or SmartFTP instead. Using FTP to transfer files allows you to transfer much larger files than are generally allowed as attachments by most ISPs. Using the third-party tool allows you to see that the file was actually transferred to the intended location. Figure 1.2 shows a connection to the FTP server at Wiley. This is one of the servers to which authors send completed work.

FIGURE 1.2 A connection to an FTP server at Wiley



Trivial File Transfer Protocol (TFTP)

TFTP is similar to FTP in that it allows the transfer of files within a network, but that's where the similarity stops. Whereas FTP allows for the browsing of files and folders on a server, TFTP requires that you know the exact name of the file you want to transfer and the exact location where to find the file. Also, whereas FTP uses the connection-oriented TCP protocol, TFTP uses the connectionless UDP protocol. TFTP is most often used for simple downloads such as transferring firmware to a network device, for example, a router or a switch.

Simple Mail Transfer Protocol (SMTP)

SMTP defines how email messages are sent between hosts on a network. You can remember SMTP as "sending mail to people." SMTP works at the Application layer of the OSI model and uses TCP to guarantee error-free delivery of messages to hosts. Since SMTP actually requires that the destination host always be available, mail systems spool the incoming mail into a user's mailbox so that the user can read it at another time. How users read the mail is determined by what protocol they use to access the SMTP server.

Hypertext Transfer Protocol (HTTP)

HTTP is the protocol that users utilize to browse the World Wide Web. HTTP clients use a browser to make special requests from an HTTP server (web server) that contains the files they need. The files on the HTTP server are formatted in web languages such as Hypertext Markup Language (HTML) and are located using a uniform resource locator (URL). The URL contains the type of request being generated (`http://`, for example), the DNS name of the server to which the request is being made, and, optionally, the path to the file on the

server. For example, if you type `http://support.microsoft.com/` in a browser, you will be directed to the Support pages on Microsoft's servers.

Hypertext Transfer Protocol Secure (HTTPS)

One of the disadvantages of using HTTP is that all the requests are sent in clear text. This means the communication is not secure and therefore unsuited for web applications such as e-commerce or exchanging sensitive or personal information through the Web. For these applications, HTTPS provides a more secure solution that uses a Secure Sockets Layer (SSL) to encrypt information that is sent between the client and the server. For HTTPS to operate, both the client and the server must support it. All the most popular browsers now support HTTPS, as do web server products such as Microsoft Internet Information Services (IIS), Apache, and most other web server applications. The URL to access a website using HTTPS and SSL starts with `https://` instead of `http://`. For example, `https://partnering.one.microsoft.com/mcp` is the page that is used to authenticate Microsoft Certified Professionals to Microsoft's private website.

Post Office Protocol Version 3 (POP3)

POP3 is one of the protocols that is used to retrieve email from SMTP servers. Using POP3, clients connect to the server, authenticate, and then download their email. Once they have downloaded their email, they can then read it. Typically, the email is then deleted from the server, although some systems hold a copy of the email for a period of time specified by an administrator. One of the drawbacks of POP3 authentication is that it is generally performed in clear text. This means that an attacker could sniff your POP3 password from the network as you enter it.

Internet Message Access Protocol Version 4 (IMAPv4)

IMAPv4 is another protocol that is used to retrieve email from SMTP servers, but IMAPv4 offers some advantages over POP3. To begin with, IMAPv4 provides a more flexible method of handling email. You can read your email on the email server and then determine what you want to download to your own PC. Since the email can stay in the mailbox on the server, you can retrieve it from any computer that you want to use, provided that the computer has the software installed to allow you to access the server. Microsoft Hotmail is a good example of an IMAPv4 type of service. You can access your Hotmail mail from any browser. You can then read, answer, and forward email without downloading the messages to the computer that you are using. This can be very convenient for users who travel.

Telnet

Telnet is a virtual terminal protocol that has been used for many years. Originally, Telnet was used to connect "dumb terminals" to mainframe computers. It was also the connection method used by earlier Unix systems. Telnet is still in use today to access and control network devices such as routers and switches.

The main problem with Telnet for today's environment is that it is not a secure protocol; everything is transmitted in plain text. For this reason, Telnet is being replaced by more secure methods such as Secure Shell and Microsoft's Remote Desktop Connection, which provide encrypted communication.

Secure Shell (SSH)

First developed by SSH Communications Security Ltd., Secure Shell is a program used to log into another computer over a network, execute commands, and move files from one computer to another. SSH provides strong authentication and secure communications over unsecure channels. It protects networks from attacks such as IP spoofing, IP source routing, and DNS spoofing. The entire login session is encrypted; therefore, it is almost impossible for an outsider to collect passwords. SSH is available for Windows, Unix, Macintosh, and Linux, and it also works with RSA authentication.

Internet Control Message Protocol (ICMP)

The ICMP protocol works at the Network layer of the OSI model and the Internet layer of the TCP/IP protocol suite. ICMP provides error checking and reporting functionality. Although ICMP provides many functions, the most commonly known is its ping utility. The ping utility is most often used for troubleshooting. In a typical "ping scenario," an administrator uses a host's command line and the ping utility to send a stream of packets called an *echo request* to another host. When the destination host receives the packets, ICMP sends back a stream of packets referred to as an *echo reply*. This confirms that the connection between the two hosts is configured properly and that the TCP/IP protocol is operational.

ICMP can also send back messages such as "Destination Host Unreachable" or "Time Exceeded." The former is sent when the host cannot be located on the network, and the latter is sent when the packets have exceeded the timeout period specified by TCP. Still another function of ICMP is the sending of source quench messages. These messages are sent by ICMP when the flow of data from the source is larger than that which can be processed properly and quickly by the destination. A source quench message tells the system to slow down and therefore prevents the resending of many data packets.

Address Resolution Protocol (ARP)

The ARP protocol works at the Network layer of the OSI model and the Internet layer of the TCP/IP suite. It is used to resolve IP addresses to MAC addresses. This is an extremely important function, since the only real physical address that a computer has is its MAC address; therefore, all communication will have to contain a MAC address before it can be delivered to the host. This is accomplished in a series of steps:

1. A computer addresses a packet to another host using an IP address.
2. Routers use the IP address to determine whether the destination address is in their network or on another network.

3. If a router determines that the address is on another network, it forwards the packet to another router based on the information that is contained in its routing table.
4. When the router that is responsible for the network that contains the destination address receives the packet, it checks the ARP cache to determine whether there is an entry that resolves the IP address to a MAC address. If there is an entry, it uses the MAC address contained in the entry to address the packet to its final destination.
5. If there is no entry in the ARP cache, the router resolves the IP address to a MAC address for the destination by using ARP to broadcast onto the local network. It asks the computer with the IP address contained in the destination address of the packet to respond with its MAC address. The router also gives the computer its own MAC address to use for the response.
6. The broadcast is "heard" by all the computers in the local network, but it will be responded to only by the computer that has the correct IP address. All other computers will process the request only to the point that they determine it is not for them.
7. The computer that is configured with the IP address in question responds with its MAC address.
8. The router addresses the packet with the MAC address and delivers it to its final destination.

Reverse Address Resolution Protocol (RARP)

RARP, as its name implies, is the opposite of ARP. RARP resolves a MAC address to an IP address. RARP was first used by diskless workstations to obtain an IP address from a server before DHCP servers were available. It simply presented its MAC address and was given an IP address based on its MAC address. RARP is sometimes used as a rudimentary form of security on applications.

Network Time Protocol (NTP)

NTP is a protocol that works at the Application layer of the OSI model and synchronizes time between computers in a network. In today's distributed networks, ensuring that the time is synchronized between clients and servers is essential. Authentication protocols, such as the Kerberos protocol used with Microsoft's Active Directory, use keys that are valid only for about five minutes. If a client and a server are not synchronized, the keys could be invalid the very second they are issued. In many of today's networks, an authoritative time source such as the Internet is first used and configured onto a time server (perhaps a domain controller). Then that server uses NTP to synchronize time with other computers in the network. Some computers may be a receiver of the correct time as well as a sender of the time to other computers in the network.

Internet Group Multicast Protocol (IGMP)

IGMP is the standard for IP multicasting on intranets. It is used to establish host memberships in multicast groups on a single network. The mechanisms of the protocol allow a host to inform its local router, using Host Membership Reports indicating that it wants to receive messages addressed to a specific multicast group.

Session Initiation Protocol (SIP)

SIP is a Session layer protocol that is primarily responsible for setting up and tearing down voice and video calls over the Internet. It also enables IP telephony networks to utilize advanced call features such as SS7.

Real-Time Transport Protocol (RTP)

RTP defines a standardized packet format for delivering audio and video over the Internet. It can also be used with other protocols, such as RTSP, to enhance the field of multimedia applications. It is frequently used in streaming, video conferencing, and push-to-talk applications.

Simple Network Management Protocol (SNMP)

The SNMP protocol is used to monitor devices on a network. A software component (called an *agent*) runs on the remote device and reports information via SNMP traps to the management systems. These management systems can be configured to record information such as errors on a network or resource information of the computers on a network.

SNMPv2 is an enhancement to the original SNMP (SNMPv1). The management information databases used in SNMPv1 are cumbersome and confusing to an administrator. SNMPv2 provides more user-friendly input and output options for data. SNMPv3 adds security measures for message integrity, authentication, and encryption. The enhancements of SNMPv3 have made the previous two versions obsolete. The RFC that defines SNMPv3 (RFC-3411) refers to the previous versions as "historic."

Transport Layer Security (TLS)

TLS allows network devices to communicate across a network while avoiding eavesdropping, tampering, and message forgery. It is designed to allow end users to be sure with whom they are communicating. Clients can negotiate the keys that will be used to secure the data to be transferred. TLS is set to supersede its predecessor SSL.

Exam Essentials

Know the purpose of each protocol in the TCP/IP protocol suite. You should understand the general purpose for each protocol in the TCP/IP protocol suite. In addition, you should understand how the protocols work together.

Know the function and use of each protocol in the TCP/IP protocol suite. You should know the function for each protocol in the TCP/IP protocol suite. In addition, you should know the level of the OSI model at which each protocol functions.

1.2 Identify commonly used TCP and UDP default ports

If people performed only one task at a time with each computer, there might not be a need for ports, but we all know that computers can perform many tasks at one time. Because this is the case, you need a way to identify packets so that they will be processed by the computer in the correct manner. By identifying each packet with a port number, you assure that the computer will direct the packet to the right area within it where the appropriate processes can be performed.

TCP and UDP port numbers are used to identify packets in regard to the services that they require. You can also filter traffic using these port numbers to restrict only specific types of traffic from a network. You should understand how TCP and UDP ports can be used to facilitate and control traffic. In the following sections, I will discuss the various types of TCP and UDP ports and describe their general use. You should be able to identify the port number that each of the most common network protocols, services, and applications use. You should know the port number when given a service as well as the service when given a port number.

Port Designations

TCP/IP has 65,536 ports available. As you can imagine, some ports are used much more than others. Ports are divided into three main groups, or designations:

Well-known ports These port numbers range from 0 to 1023. These are the most commonly used ports and have been used for the longest period of time. When CompTIA states that you should know the definition of well-known ports, it's referring to the ports in this group.

Registered ports These port numbers range from 1024 to 49151. Registered ports are used by applications or services that need to have consistent port assignments. These ports, like the well-known ports, are agreed upon by most organizations for standardization of use.

Dynamic or private ports These port addresses range from 49152 to 65535. These ports are not assigned to any particular protocol or service and can therefore be used for any service or application.

It is common for applications to establish a connection on a well-known port and then move to a dynamic port for the rest of the conversation. It's important that you understand port numbers, because you may be configuring them for communication purposes as well as to provide filtering and therefore prevent the communication of specified applications or

services. In the next section, you will examine the most common specific port assignments more closely.

Now that I have discussed the general nature of ports, I'll get much more specific in regard to well-known ports. Although there are 1,024 well-known ports, only a handful of these are commonly used on networks. The ones that are used most frequently are not arranged in any logical order in regard to their use, so unfortunately the only way to remember most of them is just to memorize them. In the following sections, I will pair up each of the most commonly used well-known ports with its protocol, service, or application.

Well-Known Port Numbers

As I said earlier, *well-known* is the name given for the port designation, but not all of the numbers between 0 and 1023 have a well-known service assigned to them. The good news is that you don't have to memorize 1,024 port assignments! Aren't you relieved? The bad news, however, is that you do have to memorize the port assignments in Table 1.2. Sorry!

TABLE 1.2 The Most Common Well-Known Port Numbers and Associated Services

Service, Protocol, or Application	Port Assignment	TCP, UDP, or Both
File Transfer Protocol (FTP)	20, 21	TCP
Secure Shell (SSH)	22	TCP
Telnet	23	TCP
Simple Mail Transfer Protocol (SMTP)	25	TCP
Domain Name System (DNS)	53	UDP
Trivial File Transfer Protocol (TFTP)	69	UDP
Hypertext Transfer Protocol (HTTP)	80	TCP/UDP
Post Office Protocol version 3 (POP3)	110	TCP
Network Time Protocol (NTP)	123	TCP
Internet Message Access Protocol version 4 (IMAP4)	143	TCP
Simple Network Management Protocol (SNMP)	161	UDP
HTTPS	443	TCP

You should memorize the port numbers in the Table 1.2 so that you can recognize them in the configuration of servers, routers, switches, and other network equipment. You might use them to configure a service or protocol. In addition, you might use them to filter a protocol or service on a firewall. In either case, a familiarity with the port numbers will assist you in configuration as well as in communication about the services and applications themselves.

Exam Essentials

Know the three port types and their ranges. You should know the three main types of ports: well-known, registered, and dynamic. In addition, you should be able to identify the ranges of each type of port. You should know when and where each port type might be used.

Know the most common well-known ports. The well-known ports should be very well known to you! You should be able to identify the most common well-known ports. You should understand the purpose of each type of port and the application or service that uses it.

1.3 Identify the following address formats: IPv6, IPv4, MAC addressing

In the end, all network devices find each other by their MAC addresses. When it comes to delivering a frame from one host to another, the next MAC address of a computer or a router interface in the path toward the client must be known. Essentially overlaid on top of the MAC address is a logical address that assists you in building complex networks. In the past, we have used primarily one protocol for this logical addressing, IPv4. In the last few years a new type of addressing, IPv6, has emerged that will allow for the growth of our industry and provide the security and control mechanisms that are needed with today's networks. You should be able to recognize IPv6, IPv4, and MAC addresses, and you should be able to differentiate between the different types of addresses. You should understand the format of each type of address and the difference between how network engineers interpret it and how network devices read it.

IPv6

You probably wouldn't think you would ever be in danger of running out of a group of things if you had 4 billion of them to start! Well, that is what happened with IPv4 addresses. Later, I will discuss the structure of IPv4 addresses, and then you can see what happened. First, I'll talk about what the world is going to do next in regard to logical addressing.

IPv6 is the latest logical addressing scheme for networks. Each IPv6 address is a 128-bit binary address represented in hexadecimal numbers. Most companies are not being "forced" into IPv6 as of yet. The latest server and client operating systems (Windows XP, Windows Vista, Windows Server 2003, Windows Server 2008, and so forth) support the protocol, but you don't necessarily have a compelling reason to change as of yet. When you do change, you will need to know a little about hexadecimal addresses to be able to interpret what you are seeing in an IPv6 address.

The following is an IPv6 address on my laptop:

fe80::218:deff:fe08:6e14

That looks pretty weird, doesn't it?

Now I'll talk about what this really says and how you should interpret it.

Each hexadecimal character in the address is actually seen by the network device as a binary number with 4 bits. Table 1.3 illustrates the relationship of each decimal, binary, and hexadecimal number and/or character.

TABLE 1.3 Decimal Binary and Hexadecimal Conversion

Decimal	Binary	Hexadecimal
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD

TABLE 1.3 Decimal Binary and Hexidecimal Cionversion *(continued)*

Decimal	Binary	Hexadecimal
14	1110	0xE
15	1111	0xF

As you can see from the table, if you were first to convert each of the characters you see in your address into its binary equivalent, the result would be as follows:

1111 1110 1000 0000 :: 0010 0001 1000 : 1101 1110 1111 1111 : 1111 1110 0000 1000 : 0110 1110 0001 0100

Not so fast, though! Equally important as what you see is what you do not see but you still know must be there. For example, you know that there are a total of 128 bits in this address. Also, you know that each section between a set of colons should actually have 16 bits on its own, so what are you missing?

Well, to begin with, any "leading zeros" can be interpreted by the device easily and can therefore be left out, as you can see in the second section of the previous address. In addition, successive fields of zeros can be represented as ::, but this can be done only once in an address, because otherwise the device wouldn't know how many successive zeros were represented by each ::. If you do a quick count, you will find that you are missing 52 zeros! In other words, although you can represent the IPv6 address in this case as the following hexadecimal number:

fe80::218:deff:fe08:6e14

what the device will use is a 128-bit number that looks like the following:

1111 1110 1000 0000 : 0000 0000 0000 0000 : 0000 0000 0000 0000 : 0000 0000 0000 0000 : 0000 0010 0001 1000 : 1101 1110 1111 1111 : 1111 1110 0000 1000 : 0110 1110 0001 0100

As you can see, this is a huge addressing system that should allow for an almost limitless supply of addresses. Of course, the last time someone said that, we soon began to run out of addresses!

IPv4

Now that you have seen the wildness of an IPv6 address, you should be glad to talk about the mundane IPv4 address again. An IPv4 address is a 32-bit binary address represented in what we call *dotted decimal format*. The following is an example of an IPv4 address:

192.168.1.1

In addition to the IP address, a subnet mask is also used with IPv4, which has the effect of "measuring" the address to determine which parts of it are the network portion and which parts are the host portion. You can think of the network portion as the street

on which you live and the host portion as the specific address of your house or apartment. Simply put, where there are 1s in the binary of the subnet mask, the corresponding bits in the IPv4 address are network bits; and where there are 0s in the binary of the subnet mask, the corresponding bits in the IPv4 address are host bits.

Now you may be thinking that IPv4 isn't in the binary form — IPv4 is in the dotted decimal format. Well, the network devices "see" the IPv4 addresses as binary numbers. In fact, 192.168.1.1 ends up looking like the following:

11000000 10101000 00000001 00000001

"How does that happen?" you may ask. Well, I'm glad you asked. The dotted decimal form uses the first 8 bits of binary over and over four times. The bits of the address are then valued based on the following template of values:

128 64 32 16 8 4 2 1 . 128 64 32 16 8 4 2 1 . 128 64 32 16 8 4 2 1 . 128 64 32 16 8 4 2 1

The address would then line up with the template as follows:

1 1 1 1 1 1 1

Everywhere there is not a 1 is a 0.

Later, I will discuss how the subnet mask combines with the IP address to determine which bits will be network bits and which will be host bits. I will also discuss how you can use a custom subnet mask to subnet a network further for more efficient and effective use of IP addresses.

MAC Addressing

Every device in a network must learn the MAC address of another device in order to communicate with it. Since we represent MAC addresses as hexadecimal numbers, it only makes sense to assume that network addresses must be able to read hexadecimal code — but you know what happens when you assume! In reality, network devices can only read binary, so the hexadecimal representation of the MAC address is in fact interpreted by the device as a binary number. The following is a MAC address on my computer:

00-18-DE-08-6E-14

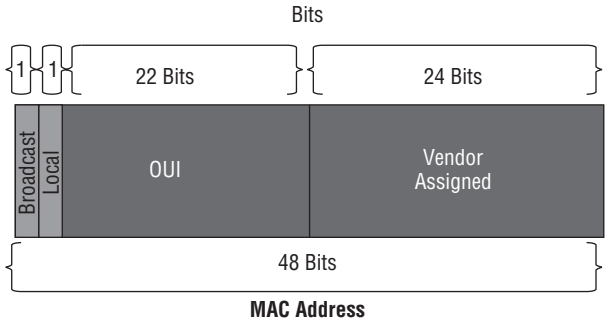
If you examine this address closely against Table 1.3, you will note that its binary equivalent is the following:

0000 0000 – 0001 1000 – 1101 1110 – 0000 1000 – 0110 1110 – 0001 0100

In other words, the MAC address is actually a 48-bit binary address that is represented as hexadecimal. Figure 1.3 illustrates the structure of a MAC address. The first two bits on the left (high order) represent whether the address is broadcast and whether it is local or remote. The next 22 bits are assigned to vendors that manufacture network devices, such as routers and network interface cards (NICs). This is called the *organizational unique identifier* (OUI). The next 24 bits should be uniquely assigned in regard to the OUI. In

other words, if I am 3COM and I have already used a specific hexadecimal number with one of my OUIs, then I should not use it again. In this way, each NIC has an address that is as unique as a person's fingerprint.

FIGURE 1.3 The structure of a MAC address



The main point to remember about MAC addresses is that they should be unique within the network in which they are to be used. This means that if one is assigned to a NIC, it should be unique within the whole world; but if a MAC addresses is functioning only on an interface within your LAN, then you should just ensure that it's unique within your LAN. Sometimes administrators may change the MAC address on a router interface, for example, to facilitate a behavior of another protocol. These types of changes are beyond the scope of this book.

Exam Essentials

Know how to identify an IPv6 address. You should know that an IPv6 address is a 128-bit binary address represented in hexadecimal. In addition, you should understand that leading 0s and successive fields of 0s may be omitted when representing an IPv6 address.

Know how to identify an IPv4 address. You should know that an IPv4 address is a 32-bit binary address that is represented in dotted decimal format. You should understand that the address is divided into four sections, which each contain 8 bits and are therefore called *octets*. In addition, you should understand how a subnet mask combines with an IP address to determine which bits are network and which are host.

Know how to identify a MAC address. Understand that a MAC address is a 48-bit binary address that is represented in hexadecimal code. You should know that MAC addresses are assigned to NICs, routers, switches, and other network equipment and should be unique in the network in which they are to be used. In addition, you should realize a MAC address must always be determined in order for communication to move from any host on a network to any other host.

1.4 Given a scenario, evaluate the proper use of the following addressing technologies and addressing schemes: Classful vs. classless, NAT, PAT, SNAT, Public vs. Private, DHCP, Unicast, Multicast, Broadcast

Today's networks are not "your father's network." Networks continue to evolve, and what we want to do on them continues to evolve. We are placing very fast computers on our networks now and expecting to receive reports, email, chat, music, videos, games, and so forth — often all at once! Because of these challenges, network administrators have to rely on newer and better technologies to both control traffic and to provide security for a network. In addition, you have to rethink some of the schemes used to push data around a network. In the following sections, I'll discuss both of these important topics. You should be able to evaluate the proper use of many different network technologies such as subnetting, supernetting, NAT, PAT, DHCP, and others. In addition, you should be able to evaluate the difference between and the proper use of unicast, broadcast, and multicast traffic.

Addressing Technologies

Today's networks use IP addressing in many creative ways based on the needs of the administrator and ultimately the needs of the users on the network. Some methods that administrators use to customize their networks include subnetting, classful addressing, classless addressing, NAT, PAT, SNAT, public addressing, private addressing, DHCP assigned addresses, static assigned addresses, and APIPA addresses. I will discuss each of the concepts in detail.

Subnetting

Subnetting is a method used to create additional broadcast domains. You may wonder why you want additional broadcast domains when broadcasts are typically considered bad; that is, they are something to be avoided whenever possible. Look at it this way: if you have a fixed number of hosts in a network, you can reduce the number of hosts per broadcast domain and therefore reduce the effect of broadcasts on the hosts by increasing the number of broadcast domains. This is because there will be fewer hosts in each of the broadcast domains.

In addition to reducing the effect of broadcasts, subnetting also allows you to apply security policies in an easy and efficient manner. Each subnet can represent a location, role, job, and so on. By applying access control lists and other types of network filtering rules,

you can control who gets access to what on a network. This job would be made much more difficult if you could not use subnets.

Now that you know the "why" of subnetting, I'll cover the "how" of subnetting. In plain terms, when you subnet IPv4, you are just reapplying the same sets of rules that were used to create the classful system of IP addressing in the first place. Because of this, it's only fitting that I begin there.

The early developers of IPv4 established a classful system of IP addresses that defined five classes of addresses. The engineers wanted to identify the type of class as quickly as possible in the addressing, so they actually did it in the first three bits of the address. Table 1.4 references how this was done and the effect it has on the number of networks and hosts per network.

TABLE 1.4 IPv4 Classful Addressing System

Class	First Octet Range	Subnet Mask	Number of Networks	Number of Hosts/ Networks
A	00000001–01111111 1–126 (127 is reserved)	255.0.0.0	126	16,777,214
B	10000000–10111111 128–191	255.255.0.0	16,384	65,534
C	11000000–11011111 192–223	255.255.255.0	2,097,152	254

As you can see in the table, when the first bit of the address was a 0 and the subnet mask was 255.0.0.0, then the address was a Class A address. There were many more Class B addresses, but they could not have a lot of hosts! These were generally assigned to the military, government, and very large corporations.

When the first bit was a 1, the second bit was a 0, and the subnet mask was 255.255.0.0, then the address was a Class B address. There were many more Class B addresses, but they can't have the tremendous number of hosts as Class A. These were generally assigned to medium-sized to large corporations and smaller governmental entities.

When the first bit was a 1, the second bit was also a 1, the third bit was a 0, and the subnet mask was 255.255.255.0, then the address was a Class C address. There were a great number of Class C addresses, but each one could contain only 254 hosts. These were originally used for small companies and very small government entities.

Now, you may have noticed that I've been speaking in the past tense. That's because we don't follow this classful system anymore, but that doesn't mean you don't need to know it! What we *do* follow is based on the classful system, but we have customized it to fit our needs using logical addressing methods and new technologies such as NAT, PAT, and proxies, which I will discuss later.

In today's networks, you need to make the most efficient use possible of the IP addressing space that you have been given by the Internet Corporation for Assigned Names and Numbers (ICANN) or that you have created for yourself with private IP addressing. To do this, you use custom subnet masks that define the appropriate number of networks and the appropriate number of hosts per network for your particular situation.

You generally start with a classful address that has the capacity to be subnetted further to meet the needs of your network. For example, let's say I have one network defined as 192.168.1.0 with a subnet mask of 255.255.255.0. As I discussed before, this subnet mask identifies the network bits and host bits in the network. If you were to convert the dotted decimal subnet mask to binary, you would find twenty-four 1s in a row followed by eight 0s in a row. This means that the network portion of the address is 192.168.1. The 0 identifies the beginning of new network, and the addresses after it would be 1 to 254. The last address would be 255; this is not a host address but rather a broadcast address. "What's the difference?" you may ask. Well, if another host wanted to address a packet in such a way that it would be received by all 254 hosts (in this case), then the host would use 192.168.1.255, which is the broadcast address. The broadcast address should be set aside for broadcasts and never be used as a host address.

Now that you have established what you already have, let's say what you have is not what you want. Let's say you want to have 8 subnets with as many hosts as possible in them instead of just one network with 254 hosts. What would you do then? You guessed it — you would subnet the classful network to create the custom networks you need. How would you do this?

You would begin by understanding that you have 8 host bits with which to work. The network bits will not be changed, and you will always be moving from the left to the right on your template. The question now is "How many of those 8 host bits do you need to change into subnet bits to create the eight subnets that you need?" (Some people refer to this part as *borrowing*, which is a term I never really liked because I'm not really planning on "giving them back.") The answer to this question lies in the formula $2^s \geq \#$ of subnets. In this formula, s is the number of host bits that will be turned into subnet bits and # *of subnets* is the number of subnets you need to create.

In this case, $2^s \geq 8$. Solving for s, you find that it must be at least 3. You want the lowest s that works because you also want to maximize the number of host bits that you still have remaining, so $s = 3$. Now the next question is "Which three?" Well, you are always going to move from the left to the right, so you will start at the left of the remaining 8 bits and take the first 3 bits from the left toward the right. This means that the subnet bits will be the 128, 64, and 32 bits. To make these host bits into subnet bits, you will simply change the corresponding bits in the subnet mask from 0 to 1. When you make this change, the subnet mask will then change to 255.255.255.224 since 128 + 64 + 32 = 224.

The next question on your mind might be "Then what are my 8 subnets?" You can answer this question by determining the increment of the subnets and therefore their numbers and ranges of hosts. The increment is always 256 — the last number in the subnet mask that is not a 0. In this case, it's 256 – 224, which equals 32. The first network is always the same as what you started with, but with a new subnet mask. You can express the new subnet mask as 255.255.255.224, or you can express it by using a forward slash at the end of

the IP address followed by a number indicating the number of 1s in the subnet mask. In this case, you could express your subnet mask as a /27. This is referred to as *CIDR notation*.

Since all the other networks are determined by the increment, your networks will be as follows:

192.168.1.0/27

192.168.1.32/27

192.168.1.64/27

192.168.1.96/27

192.168.1.128/27

192.168.1.160/27

192.168.1.192/27

192.168.1.224/27

The host ranges and broadcast addresses can then be determined without any further use of the binary. For example, the 0 network will have 30 hosts in it ranging from 1 to 30, and it will have a broadcast address of 31. The 32 network hosts will range from 33 to 62 with a broadcast address of 63, and so on, through the networks.

You can also check your math by understanding that the number of hosts will always be $2^h - 2$, where h is the number of remaining host bits after the subnet bits are determined. In this example, there are five remaining host bits, so the formula will be $2^5 - 2 = 30$. Since this matches the number of hosts as determined by the increment, you know you are on the right track!

Now let's try one that is a little more complicated. Don't worry, I'll walk you right through it, and then you will be able to do it yourself. Let's say you have an IP network of 172.16.0.0 with a subnet mask of 255.255.0.0 and you want to have 60 subnets with as many hosts per subnet as possible. What would the new subnet mask be? How many hosts would you have? What would your networks look like?

You start solving this problem in the same way as the last by noticing where you are beginning in the address, based on the subnet mask. In other words, my first question is always "Where am I?" Since you have a subnet mask here of 255.255.0.0, you are halfway through the address. In other words, you have sixteen 1s followed by sixteen 0s in the subnet mask. The fact that you have sixteen 0s means you have 16 host bits, some of which will be used for subnet bits. The next question is "How many host bits do you need to convert to subnet bits to create the 60 subnets that you need?"

You can answer this question with the same formula as before, $2^s \geq$ # of subnets. In this case, $2^s \geq 60$. Solving for s, you determine that $s = 6$, since $2^6 = 64$, and that's the first number that is higher than 60. Now the question is "Which six?" Remember that you are always moving from left to right, so the six bits that you will use will be the first six in the third octet starting from the left. This means you will change the corresponding bits in the subnet mask from 0s to 1s. This in turn means that the subnet mask number will change to 255.255.252.0, since 128 + 64 + 32 + 16 + 8 + 4 = 252. In other words, when you change the subnet bits to 1s, the values count and change the subnet mask accordingly.

The next question is "How many hosts could you have per network?" A close look at the template should show you that you have 2 host bits left in the third octet and 8 host bits left in the fourth octet. That's a total of 10 host bits. This means you can have $2^{10} - 2$ hosts per subnet, or 1,022.

Now you might be wondering how you are going to do that and what the addresses are going to look like when you get done. Just as before, the first network is always the same network you started with, but it has the new subnet mask, and the rest of the networks are determined by the increment. In this case, your first network is 172.16.0.0 with a subnet mask of 255.255.252.0. The increment is always 256 — the last number in the subnet mask that is not a 0. In this case, the increment is 256 – 252 = 4. This means that the first three networks will be as follows:

172.16.0.0/22

172.16.4.0/22

172.16.8.0/22

Notice that I left some blank space between the network addresses. I like to call that space "thinking room," because you are going to do a lot of thinking in there. It's rather straightforward to see that the first host in the 172.16.0.0 network will be 172.16.0.1, but where do you go from there to get 1,022 hosts? Imagine an old odometer that actually spins out the 10ths of miles. Do you have that in your mind? Now when it gets to nine 10ths, think about what happens. The 10ths will then go back to 0, the number on the left will increment by 1, and then it all starts over again. Right? That's the same thing that happens with the IP addresses, except that it's not 0 to 9 but rather 0 to 255. In this case, when the addresses get to 172.16.0.255, the next number is then 172.16.1.0. Now, here's the kicker: both of those addresses are valid hosts! In fact, there will be a lot of weird-looking numbers that will be valid hosts as well. So, what is the last host in the 172.16.0.0/22 network? The last host is 172.16.3.254, and the broadcast is 172.16.3.255. After that, the 172.16.4.0/22 network starts, which has a broadcast address of 172.16.7.255. Use the "thinking room," and you will see it.

It's extremely important with today's networks that you understand IP addressing and subnetting. The quicker you can determine the subnet on which a host resides, the better you will be at network troubleshooting. I hope this has helped you to see IP addresses for what they are without having to convert them to binary numbers. With practice, you will be able to "see" the answers instead of always having to figure them out. I highly recommend you spend some time working on IP address subnetting. One tool that I've found invaluable is the website: `http://subnetttingquestions.com`. It was created in part by Todd Lammle, a fellow Wiley author. This site is free and offers hundreds of questions and answers. Your challenge is to get the same answer as the site has and to do it as quickly as possible.

Classful vs. Classless

Now that you know about the subnet mask, I can talk about classful addressing vs. classless addressing. The first thing to remember is that the names for these can throw you off

track if you aren't careful. Logically, it might seem that classful would be better than classless. However, this is not true in this case, and it isn't true that classless is always better than classful. It depends on what you are trying to accomplish in your network.

Classful addressing takes its name because the first octet of the address determines the subnet mask that will be used, and therefore the subnet mask does not have to be, and is not, advertised by the routers in the routing protocols. In other words, referring to the information in Table 1.5, you will notice that an address that has 1 to 126 in the first octet would be considered a Class A address if it had a subnet mask of 255.0.0.0. With classful addressing, that's its only choice. In other words, with classful addressing, the subnet mask is always assumed to be the one that corresponds with its first octet address. This has the affect of limiting some network designs that otherwise could have used, for example, networks 172.16.1.0 and 172.16.2.0 with other networks between them. This cannot be done because the classful routing protocols will assume both of the networks to be 172.16.0.0, because of the assumed mask of 255.255.0.0. This will result in a network scheme that will not function properly.

Now let's say you have a routing protocol that actually takes into account the address and subnet mask you assigned to the interface. Wouldn't that be nice? In that case, you could specify the networks 172.16.1.0/24 and 172.16.2.0/24 by assigning the subnet mask of 255.255.255.0 to each, rather than the classful subnet mask of 255.255.0.0. If the protocol could advertise the address along with the subnet mask, then you could use these two networks even if you had other networks between them because they would be seen as two unique networks. This type of addressing is used in today's networks because it allows for more complex networking schemes that can make more efficient use of the available IP addresses.

Network Address Translation (NAT)

NAT is a service that translates one set of IP addresses to another set of IP addresses. NAT is most often used between a private network and the Internet, but it can also be used in other ways such as to translate a group of global internal addresses to a group of global external addresses. NAT is a service that can be run on a computer, a router, or a specialized device that provides only network address translation.

Port Address Translation (PAT)

PAT is a service that most people actually think of as NAT. When you have two or more computers on the inside of a network that share one address as represented on the outside of the network (usually the outside interface address of the router), the only way to keep their network communication channels separate and organized is by port designation on each packet. PAT changes the source address of a packet as it passes through the router or other device using PAT, appending it with a specific port number. It then keeps a record of the port numbers to which it has assigned packets and the true inside local address of the computers that generated them. In this way, PAT uses ports to provide address translation for many inside source addresses to one outside source address.

Secure Network Address Translation (SNAT)

SNAT is the simplest form of NAT and is often used in conjunction with other more sophisticated forms such as PAT. It provides for a one to one translation of an inside local address (on the inside of a network) to an inside global address (on the outside of a network). One of the advantages of SNAT is that the address that will be used on the outside is completely configured and therefore very easy to determine and to troubleshoot. SNAT is often used when specialized equipment attached to a network requires a specified IP address range that is different than the organization has available.

Public vs. Private Addresses

Unique IP address assignment on the Internet was originally the responsibility of the Internet Assigned Numbers Authority (IANA), but it has been handed over to other organizations that coordinate with each other to make sure that addresses are unique. The current three major organizations for the entire world are divided geographically as follows:

- *American Registry for Internet Numbers (ARIN)*: Serves the North American continent and parts of the Caribbean
- *Asia Pacific Network Information Centre (APNIC)*: Serves the Asia Pacific Region
- *Réseaux IP Européens Network Coordination Centre (RIPE NCC)*: Serves Europe, the Middle East, and parts of Africa

Addresses that are assigned by these authorities are referred to as registered, or *public*, addresses. If you are connecting a computer to the Internet, then you must use an address that has been assigned by one of these authorities. Now I know what you are thinking: "I'm connected to the Internet, and I never contacted any of those organizations." That's probably because you use an address that is provided by your Internet service provider (ISP) that obtained the address from one of these authorities. ISPs have large blocks of IP addresses that they can assign to their clients, thereby giving them a valid and unique IP address to use on the Internet. Some large organizations still go through the process of registering for their own address blocks, but most individuals and smaller organizations simply get whatever addresses they need from their ISP.

Private IP addresses are completely different. To understand a network diagram, you have to be able to see the difference between public and private addresses. Public addresses are said to be routable, whereas private addresses are said to be nonroutable. What does this really mean? Is there something wrong with the bits in the private IP addresses that prevent them from being routed? No, the private addresses are actually nonroutable because they are filtered by the routers that would take you from one network to another on the Internet.

But now you may be asking "How do they know which addresses to filter?" Well, the original designers of the Internet set aside some groups of IP addresses to be used for private addressing. That way, even if two companies were to choose the same addresses, and even if neither of them used a firewall, there still could be no conflict because the addresses would never "see" each other. Table 1.5 lists the addresses that are automatically filtered by routers leading onto the Internet.

TABLE 1.5 Private IP Address Ranges

Class	Address Range	Default Subnet Mask
A	10.0.0.0–10.255.255.255	255.0.0.0
B	172.16.0.0–172.31.255.255	255.255.0.0
C	192.168.0.0–192.168.255.255	255.255.255.0

You may have noticed that 127 is missing. This is because the 127 network is reserved for diagnostics and testing. The most notable address on this network is the loopback address 127.0.0.1, which I will discuss in later chapters. Also, note that Class D addresses are reserved for multicasts and that Class E addresses are reserved for experimentation and future development.

As always, the full address is indicated by the IP address combined with the subnet mask. The important point to remember here is that these are the addresses that are filtered. In reality, you could use any address that you chose for the private IP addressing schemes of your network. However, if we both decided to use a public address on the inside, for example 14.1.1.1 for a router, then we could possibly see each other and have an address conflict if everything went wrong with the firewalls and other network protection. In other words, we would not be able to rely on the automatic filters throughout the Internet. This is why it is recommended to use the private IP addresses that I have listed, and this is why you should know them.

DHCP (Static, Dynamic, APIPA)

As discussed earlier, DHCP stands for Dynamic Host Configuration Protocol, but what exactly does that mean? It means you can offload a whole lot of work configuring IP addresses, subnet masks, default gateways, DNS server addresses, and much more to a server that is relatively easy to set up and maintain. Figure 1.4 shows the DHCP server tool in Windows Server 2003. Programs like this one can be used to configure addresses on client computers in a network.

In general, computer clients should obtain their IP addresses from a DHCP server whenever possible. In contrast, devices such as servers, network printers, plotters, and router interfaces should be statically configured so their addresses do not change. Figure 1.5 shows an example of a static configuration on a Windows Server 2003 server.

All client computers since Windows 98 are configured by default to obtain their IP address from a DHCP server. What if a DHCP server is not available? In that case, they are also configured by default to use an address in the range of 169.254.0.1 to 169.254.255.254. These addresses are called Automatic Private Internet Protocol Addressing (APIPA) addresses. The advantage of using APIPA is that the clients in the same network segment that could not obtain a true IP address from a DHCP server can still communicate with each other.

The disadvantage is that the clients can communicate with each other but not with the true network. This can lead to some wild troubleshooting for the "unseasoned" administrator. The bottom line is that when you see an address that begins with 169.254, you can rest assured that it was not obtained from any DHCP server!

FIGURE 1.4 The DHCP server tool in Windows Server 2003

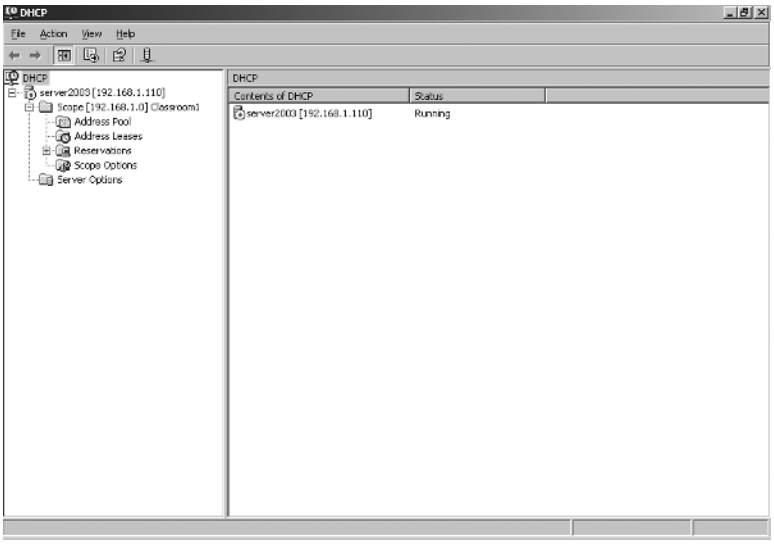
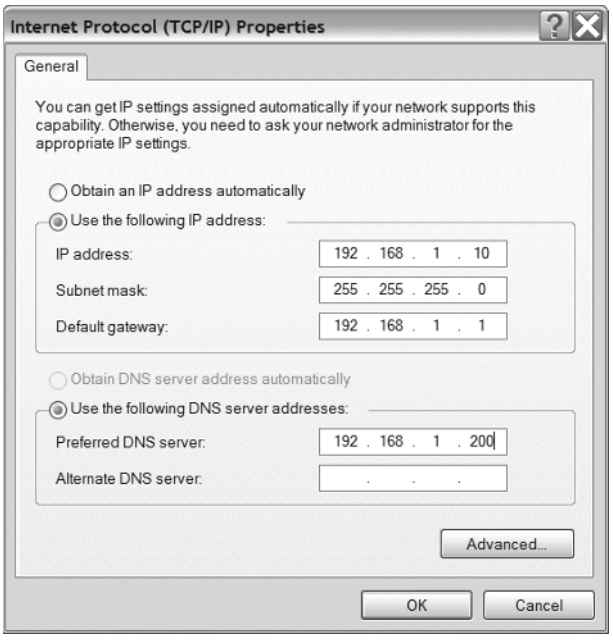


FIGURE 1.5 A static IP configuration on a DHCP server



Addressing Schemes

Three major types of addressing schemes are used on IPv4 networks. These are unicast, multicast, and broadcast. Each type has its own place in the network. In the following sections, I'll discuss each of these types of network addressing schemes.

Unicast

Of the three types of addressing schemes used in IPv4, *unicast* is the most simple and straightforward. A packet (layer 3) or frame (layer 2) is said to have a unicast address if it has one source address and one destination address. If we are discussing packets, then the source and destination addresses are of a layer 3 protocol, likely IP. If we are discussing Ethernet frames, then the source and destination addresses of a layer 2 protocol are MAC addresses. In either case, the devices need to determine only the correct unique destination address to send the packet.

Multicast

Multicast addressing can be much more complex than unicast. With multicast addressing there is still only one source address; however, there can be multiple destination addresses. In other words, the frame or packet basically carries a list of destination addresses with it, and each device checks to see whether it is on the list when it sees the data. Multicasting is especially useful for applications that send voice and video through network systems. Multicast addressing uses specialized protocols such as Internet Group Multicast Protocol (IGMP) to create and carry the "list." The IP addresses carried by IGMP can be mapped to MAC addresses for Layer 2 multicasting.

Broadcast

Broadcast addressing is similar to just standing in a room yelling out a person's name or an announcement. Anyone in the room who hears you with the name you yelled would likely respond, but everyone in the room would be disturbed in the process. On the other hand, if the announcement were actually intended for everyone in the room, then you would have accomplished your goal.

Broadcasting is accomplished by using an address that directs the data to all the members of a network or subnet. Every IPv4 network or subnet has a broadcast address, which is the last numerical address before the next network. In the binary form of a broadcast address, you will notice that all the host bits are 1s. For example, the broadcast address of the network 192.168.1.0/27 is 192.168.1.31. As you can see, the host address portion is 31 in dotted decimal, which is 11111 in binary.

Some services in an IPv4 network work by broadcasts, such as DHCP and even ARP. That said, broadcasts are typically thought of as bad and to be avoided whenever possible. IPv6 uses a different form of addressing referred to as *anycast* to avoid using broadcasts. This is beyond the scope of this chapter and not listed as an objective on the current exam.

Exam Essentials

Know the most common addressing technologies. You should know the most common addressing technologies used in today's networks, such as subnetting, NAT, PAT, DHCP, APIPA, and so on. You should understand how network engineers communicate about these technologies and how the devices use them to send data. In addition, you should understand the appropriate use of each technology in a network.

Know the most common addressing schemes. You should know the most common addressing schemes, such as unicast, multicast, and broadcast addressing. You should understand how each may be used in a network and where each one is appropriate. In addition, you should understand the underlying addressing structure in respect to the source and destination addresses of each.

1.5 Identify common IPv4 and IPv6 routing protocols

There is a big difference between a routed protocol and a *routing protocol*. In most networks today, the only routed protocol used to transmit user data from network to network is IP. It is the underlying protocol that is used by the routers to identify and accurately deliver data packets.

Today's routers determine what do with a packet that does not belong on a network to which they are connected by using a routing table. The protocols that routers use to communicate with each other about networks, and thereby build the routing table, are referred to as *routing protocols*. In the following sections, I will identify two categories of routing protocols: link state and distance vector. I will then discuss the most common protocols in each of these categories. You should understand the basic differences in the way that each routing protocol functions.

This is only a brief discussion for the purposes of the exam, because many entire books have been written about each of these protocols and their configuration.

Link State

Link state identifies and describes one of the most common categories of routing protocols in use today. *Link* means interface, and *state* means the attributes of the interface, in other words, where it is, what is connected to it, how fast it is, and so forth. Link state routing protocols send all this interface information out in the form of link state advertisements (LSAs). From these LSAs, the routers will build a map of the network. Each router in the

same area will have the same map and will therefore be able to make decisions as to how to forward a packet. The two most common link state routing protocols are OSPF and IS-IS.

Open Shortest Path First (OSPF)

OSPF is by far the most common link state routing protocol in use today. OSPF is so named because it is an "open" protocol. In other words, it's not proprietary, and it uses the Shortest Path First (SPF) algorithm developed by Dijkstra.

The principle advantages of this protocol include that it is quiet on the network — not "chatty" like some of the protocols that preceded it — and that it converges very rapidly when there is a change in the network. In other words, when the tables need to be changed to control network traffic, it makes that happen very fast — usually within a few seconds. Because of these advantages, OSPF can be used on small, medium, and large networks.

Intermediate System to Intermediate System (IS-IS)

IS-IS is another link state routing protocol that is not as popular with commercial and government networks as OSPF. Developed by the Digital Equipment Corporation, it has been used in the past mainly by large service providers. IS-IS uses Dijkstra's algorithm to make decisions about where to forward a packet, but it also uses a complex system of levels to obtain a network topology.

Distance Vector

Distance vector routing protocols are also exactly what they say they are. *Distance*, as you know, is "how far." *Vector*, as you may know, is "which direction." Distance vector routing protocols make decisions by examining these two factors against their routing tables. I will now briefly discuss each of the most common distance vector routing protocols.

Routing Information Protocol (RIP)

RIP is one of the first routing protocols. As you can imagine, being first in regard to technology does not necessarily mean being the best. In fact, RIP is now considered obsolete and is being replaced by more sophisticated routing protocols, such as RIPv2, OSPF, and IS-IS.

The principal reasons for RIP's demise are that it is a "chatty" protocol in which all information that each router knows regarding networks is broadcast every 30 seconds. In addition, RIP uses a "hop count" metric that doesn't take into account the bandwidth of a connection. Finally, RIPv1, commonly referred to as RIP, is classful, which means it does not provide the means to advertise the true subnet mask of a network. In today's varied networks, this type of routing protocol does not have the intelligence needed to route packets efficiently.

Routing Information Protocol Version 2 (RIPv2)

RIPv2 solves some of the problems associated with RIPv1, but not all of them. It does not broadcast every 30 seconds but instead uses multicast addressing for its advertisements. This provides for much more efficient use of network bandwidth. In addition, it can be

configured to be classless, which means it can carry the true subnet mask of a network and can therefore be used on more complex networks.

RIPv2, however, still uses only a hop count metric. Because of this limitation, it cannot be used effectively in today's networks that provide redundant and sometimes varied speed connections from point to point. It is therefore also considered by today's standards to be a legacy routing protocol.

Border Gateway Protocol (BGP)

BGP can be considered to be a distance vector routing protocol from autonomous system to autonomous system. An *autonomous system* is a group of devices that are under the same administrative domain, in other words, a group of devices that are under the same management and control, regardless of where they are physically located. When you connect to the Internet, you are moving from one autonomous system to another. BGP is the protocol that provides these logical connections or paths, and it is therefore also considered to be a path vector protocol. A detailed discussion of BGP is far beyond the scope of this book and is not an objective on the exam.

Hybrid

There is only one hybrid routing protocol with which you must be familiar, EIGRP. It is said to be a hybrid because it is actually a distance vector routing protocol that works like a link state routing protocol.

Enhanced Interior Gateway Routing Protocol (EIGRP)

EIGRP is a Cisco proprietary protocol that combines the ease of configuration of distance vector routing protocols with the advanced features and fast convergence of link state protocols. It is said to be a distance vector routing protocol with link state attributes. It can also be considered an advanced distance vector routing protocol or a hybrid routing protocol.

EIGRP uses a much more sophisticated metric than RIP or RIPv2. This metric includes the bandwidth of a connection and the delay, which is an experiential factor of how long it takes to pass traffic over the path of the network. It can also be "tweaked" by an administrator with load and reliability factors. Because of its more sophisticated metric, EIGRP is well suited for small, medium, and even large networks. The only possible disadvantage to EIGRP is that it is Cisco proprietary and therefore operates only on Cisco routers and Cisco layer 3 switches.

Exam Essentials

Know the most common link state routing protocols. You should be able to identify OSPF and IS-IS as the most common link state routing protocols. In addition, you should know the basic manner of function of a link state routing protocol and how it differs from distance vector routing protocols.

Know the most common distance vector routing protocols. You should be able to identify RIP, RIPv2, and BGP as distance vector routing protocols. In addition, you should understand that RIP and RIPv2 are considered to be less sophisticated because of their limited metric of hop count. Finally, you should understand that BGP is a special type of distance vector routing protocol that maps autonomous systems together, as I will discuss further in the next section.

1.6 Explain the purpose and properties of routing

Generally speaking, routers are very specialized computers that do only two things. They deliver a packet to a host that is determined to be on one of their networks, or they consult their routing tables and follow the directions there. In the following sections, I'll discuss different types of routing and routing terminology used in today's networks. In addition, I'll discuss routing tables and how routers use them to make decisions.

Interior Gateway Protocol (IGP) vs. Exterior Gateway Protocol (EGP)

All the routing protocols I've discussed thus far, with the exception of BGP, have been IGPs. BGP is an EGP. Understanding the difference relies upon your knowledge of an autonomous system. As I mentioned earlier, an autonomous system is a group of devices under the same administrative domain. If a routing protocol works within one autonomous system, it is considered to be an IGP. If it works across autonomous systems, in effect connecting them, then it is considered to be an EGP. That's all there is to it, so don't make it any harder than it really is. The only EGP that you should be concerned with today is BGP; all of the rest are IGPs.

Static vs. Dynamic

In regard to routing configuration, the term *static* means configured by the administrator. All routers could be configured with static routes alone, but that would be the hard way. Not only would it be more work initially, but every time anything changed, every route would have to be reevaluated and possibly changed as well.

Dynamic routing refers to letting the routers and routing protocols do the work for you. As I discussed earlier, there are many different routing protocols, but all of them have one element in common. They exchange information about possible paths through the network so that they can each make the best decision as to which way to send a data packet. Some routing protocols do this better than others, but they are all more efficient than strictly static routing.

Next Hop

Generally speaking, routers couldn't care less where a packet comes from when they make a routing decision. What they care about is where the packet wants to go. In other words, they are concerned with the destination address in the header of the packet. Based on the destination address, they can determine whether they can deliver the packet themselves or whether they need to send it to another router. If they cannot deliver the packet themselves, then they will consult their routing table to determine the next step. As I mentioned earlier, the routing table will give them the information as to the next interface that they can get to, which would be the appropriate place to send the packet. This interface is referred to as the *next hop* interface. This is because going from one network to another is like hopping over a router in the network diagram. It's really just going through two consecutive interfaces, but it's a lot more fun to say *hop*!

Understanding Routing Tables and How They Pertain to Path Selection

While I'm discussing routing tables anyway, I may as well show you a close-up look so you can see how it functions and how the router makes the decision by consulting it. Table 1.6 is a simple illustration of a RIPv2 route using hop count. This is actually a *Reader's Digest* version of what you might see in a Cisco router, but you get the point. As you can see, the router that contains this table knows how to get to other networks by virtue of the table. In other words, a packet that comes into this router that is destined for the 10.1.0.0 network will be sent out of a different interface from one that is destined for the 192.16.1.0 network.

TABLE 1.6 RIPv2 Hop Count

Destination Network	Subnet Mask	Interface	Metric (Hop Count)
10.1.0.0	255.255.0.0	S0	1
192.16.1.0	255.255.255.0	S1	1
172.16.0.0	255.255.0.0	S1	2

Convergence

Simply put, *convergence* means that everything is in agreement again after change has taken place. In other words, let's say you have a network that is all settled and in a *steady state*. All routers know the best interface to send a packet out based on the destination address of the packet. Now let's say you add a new interface to a router and thereby create a new path on which traffic could flow. This would cause the routing protocols to acknowledge and examine the new path and determine whether it is a more efficient path than the one they

are currently using. In fact, each router would need to examine the new path against its current path for each network in its table. It would then make a decision as to whether to make a change. This can temporarily create quite a flurry of activity on a network in regard to routing protocol information exchange.

Once all the options are considered and the decisions are made, then the activity will settle down again. A network that has settled back down is said to be have *converged*, so the process of moving through this unsettled state to the settled state is referred to as *convergence*. Some routing protocols offer much faster convergence than others. As I discussed earlier, routing protocols such as EIGRP and OSPF are "smarter" and thus are not normally chatty, but they become very chatty for a short burst of time when something changes on the network. Their ability to move very quickly from an unsettled state to a settled state is referred to as *fast convergence*. This means that a change on an interface that affects the routing tables will have minimal effect on the user data that is traversing the network.

Exam Essentials

Know the difference between IGP and EGP. You should know most routing protocols are IGP and that they work within one autonomous system. In addition, you should understand that BGP is the only EGP in common use today and that it works between autonomous systems, connecting the Internet.

Know the difference between static and dynamic routing. You should understand that static routes are those configured manually by a network administrator. In addition, you should know that static routes have their limitations because all routes might have to be reevaluated based on any change to any router interface. You should also realize that dynamic routes are preferable to static routes because the routers do the work, and it can be done very quickly.

Know the concepts of routing tables, next hop, and convergence. You should know that routers use routing tables to make decisions as to the interface to use for each network. In addition, you should understand that routers are simply looking for the next hop for any traffic that is not destined to one of their directly connected networks. You should also comprehend the concept of convergence in a network and the value of using a routing protocol with fast convergence.

1.7 Compare the characteristics of wireless communication standards

Over the past 10 years or so, wireless communication has continued to grow in business as well as in home networks. Teams of engineers have developed many standards, identified by the IEEE number 802.11 and a letter (such as *g*), to make wireless communications faster and more secure. In the following sections, I'll discuss the most common of these standards and how they are used in today's networks.

802.11 a/b/g/n

802.11 is the IEEE specification that is used for wireless LAN technology. 802.11 specifies an over-the-air interface between a wireless client and a base station or between two wireless clients. The IEEE accepted the specification in 1997. The original 802.11 standard used a frequency hopping spread spectrum radio (FHSS) signal. There have been many revisions to the standard since then. The following are the major 802.11 standards in use today:

802.11a Uses orthogonal frequency division multiplexing to increase bandwidth. This standard uses the 5GHz radio band and can transmit at up to 54Mbps. It is not widely used today.

802.11b Uses direct sequence spread spectrum (DSSS) in the 2.4GHz radio band. This standard can transmit at up to 11Mbps with fallback rates of 5.5Mbps, 2Mbps, and 1Mbps. It is one of the most commonly used standards today.

802.11g Uses DSSS and the 2.4GHz radio band. This standard enhances the 802.11b standard and can transmit at speeds up to 54Mbps. It is one of the most commonly used standards and is backward compatible to 802.11b, since they both can use DSSS.

802.11n Uses DSSS and the 2.4GHz radio band. This standard enhances the 802.11g standard and can transmit at speeds up to 600Mbps, although most devices in use today support speeds only up to about 300Mbps. This is not commonly used yet but is available and is backward compatible to 802.11g and 802.11b.

Authentication and Encryption

In the past, it was hard to say *wireless* and *security* in the same sentence without smiling a little at the irony. Gradually, newer technologies have surfaced that are slowly making these two concepts compatible with each other. In the next sections, I'll first discuss the earlier protocols that were not very secure in regard to authentication and encryption and then move on to the latest protocols that do offer some security for wireless communications.

Wired Equivalent Privacy (WEP)

One of the first attempts at wireless security was Wired Equivalent Privacy (WEP), which attempted to secure wireless connections on 802.11b-based networks. WEP attempted to secure the connections by encrypting the data transfer, but WEP was found not to be equivalent to wired security because the security mechanisms that were used to establish the encryption were not encrypted. WEP also operates only at the lower layers of the OSI model and therefore cannot offer end-to-end security for applications. Because of these shortcomings, many people have chosen newer and more sophisticated methods of securing wireless communications.

Wi-Fi Protected Access (WPA)

Wi-Fi Protected Access (WPA) was designed to improve upon WEP as a means of securing wireless communications. It can usually be installed as an upgrade on systems that currently use WEP. WPA offers two distinct advantages over WEP:

- Improved data encryption through the Temporal Key Integrity Protocol (TKIP), which scrambles the keys using a hashing algorithm. TKIP also provides an integrity-checking feature that ensures that the keys haven't been tampered with or altered.
- User authentication through the use of the Extensible Authentication Protocol (EAP) and user certificates. This ensures that only authorized users are given access to the network.

802.1x

The latest and most advanced form of wireless security is 802.1x, which is the name for the IEEE standard it supports. This type of wireless security is a standard feature of the latest operating systems such as Windows XP Professional. Access can be controlled per user and per port. 802.1x can use EAP to provide the following methods of authentication:

EAP Transport Level Security (EAP-TLS) This is the strongest method of encryption. EAP-TLS requires a certificate-based security environment. In other words, a form of certificate authority must be used. It provides mutual authentication, negotiation of the encryption method, and encrypted key determination between the client and the authenticator.

Protected EAP (PEAP) PEAP uses TLS to enhance the security of other authentication methods, such as CHAP and others. PEAP can be used without certificates unless it is being used in conjunction with MS-CHAP v2, which requires certificates in order to provide mutual authentication between the client and the server.

RADIUS

Using Remote Authentication Dial-In User Services (RADIUS), clients can be authenticated to use a wireless connection based on a current logon that can be authenticated by a domain controller. This method is used only when the user has an account in a domain such as a Microsoft Windows Active Directory domain.

Review Questions

1. What is name of the unique physical address that is assigned to every network interface card?
   - **A.** IP address
   - **B.** Hostname
   - **C.** MAC address
   - **D.** NetBIOS name
2. How many bits are used to create an IPv4 address?
   - **A.** 8
   - **B.** 6
   - **C.** 32
   - **D.** 64
3. If you have a Class B address with a default subnet mask and you need to create eight subnets, then which of the following subnet masks should you use?
   - **A.** 255.255.255.240
   - **B.** 255.255.224.0
   - **C.** 255.255.240.0
   - **D.** 255.240.0.0
4. Which of the following IP addresses are valid only for private IP addressing that is filtered from the Internet? (Choose two.)
   - **A.** 10.1.1.1
   - **B.** 172.17.255.254
   - **C.** 11.1.2.4
   - **D.** 193.168.2.1
5. Which information directory protocol is the standard for file transfer over the Internet?
   - **A.** TCP
   - **B.** UDP
   - **C.** FTP
   - **D.** HTTP
6. Which of the following are designated as well-known port numbers? (Choose two.)
   - **A.** 80
   - **B.** 49150
   - **C.** 1011
   - **D.** 8080

7. Which wireless protocol was designed to improve upon WEP and can be installed as an upgrade to WEP in many instances?
   - **A.** 802.1x
   - **B.** Kerberos
   - **C.** WAP
   - **D.** WPA

8. Which of the following is a Session layer protocol that is primarily responsible for setting up and tearing down voice and video calls over the Internet?
   - **A.** SIP
   - **B.** RTP
   - **C.** HTTP
   - **D.** FTP

9. Which of the following is an example of an IPv6 address?
   - **A.** 192.168.1.1
   - **B.** C0-FF-EE-CO-FF-EE
   - **C.** fe80::216:deff:ee09:6d13
   - **D.** 255.255.240.0

10. Which of the following subnet masks should you use to obtain 100 subnets from a Class B network?
    - **A.** 255.255.254.0
    - **B.** 255.254.0.0
    - **C.** 255.255.255.128
    - **D.** There isn't enough information to answer the question.

Answers to Review Questions

1. C. A media access control (MAC) address is a unique physical address that is assigned to each network interface card. MAC addresses are "burned in" to the card at the manufacturer.

2. C. An IPv4 address is a 32-bit address. It is composed of four sections of 8 bits each, called octets. Each octet is converted to decimal for configuration purposes, but the computer uses the entire 32-bit address for communication.

3. B. If you have a Class B address with a default subnet mask, then the current subnet mask is 255.255.0.0. This means you have 16 bits for networks and 16 bits for hosts. If you want to create eight subnets, then you need to solve for $2^s > 8$. Solving for *s*, you can determine that you need to use the first 3 bits from the network address to create the subnets. The values of the first 3 bits total 224 (128 + 64 + 32), so the new subnet mask is 255.255.224.0. (This is all assuming you use the new method, which uses the first and last subnets.)

4. A, B. The valid private address ranges include the following:
   - 192.168.0.0–192.168.255.255
   - 172.16.0.0–172.31.255.255
   - 10.0.0.0–10.255.255.255

   Only answers A and B fall into these ranges.

5. C. FTP is an Application layer protocol that uses TCP ports 20 and 21. It is the standard protocol used for file transfer over the Internet.

6. A, C. The port numbers that are designated as well-known port number are those in the range of 0 to 1023. Ports in the range of 1024 to 49151 are designated as registered ports. Ports in the range of 49152 to 65535 are designated as dynamic or private ports.

7. D. Wi-Fi Protected Access (WPA) was designed to improve upon WEP. It provides improved data encryption as well as improved authentication mechanisms.

8. A. Session Initiation Protocol (SIP) is a Session layer protocol that is primarily responsible for setting up and tearing down voice and video calls over the Internet. It also enables IP telephony networks to utilize advanced call features such as SS7.

9. C. An IPv6 address is a 128-bit binary address that is expressed as a hexadecimal address. Leading 0s can be omitted and successive fields of 0s can be represented as ::.

10. A. A Class B network will by definition have a subnet mask of 255.255.0.0. This means you have 16 host bits with which to work to create 100 subnets. To determine the number of subnets you need, you use the formula $2^s \geq 100$. Solving for s, you can determine that you need 7 host bits, since $2^6 = 64$ but $2^7 = 128$. This means that the 128, 64, 32, 16, 8, 4, and 2 values in the third octet will now count in the subnet mask. Adding these values, you get 254, so the new subnet mask is 254.