ePUB is an open, industry-standard format for eBooks. However, support of ePUB and its many features varies across reading devices and applications. Use your device or app settings to customize the presentation to your liking. Settings that you can customize often include font, font size, single or double column, landscape or portrait mode, and figures that you can click or tap to enlarge. For additional information about the settings and features on your reading device or app, visit the device manufacturer's Web site. Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the eBook in single-column, landscape mode and adjust the font size to the smallest setting. In addition to presenting code and configurations in the reflowable text format, we have included images of the code that mimic the presentation found in the print book; therefore, where the reflowable format may compromise the presentation of the code listing, you will see a "Click here to view code image" link. Click the link to view the print-fidelity code image. To return to the previous page viewed, click the Back button on your device or app.

Thanks are due to Eleanor (Ellie) Bru for working on this title once more and making it as strong as it can be. An enormous amount of credit for this book goes to Chris Crayton, without whom this edition would be only a shadow of what it is. It was an honor to work with him again, and I owe him enormous gratitude. Thanks continue to be due to Mike Harwood, who wrote the first few editions, and to the team of talented individuals at Pearson who work behind the scenes and make each title the best it can be.

Chris Crayton (MCSE) is an author, technical consultant, and trainer. In the past, he has worked as a computer technology and networking instructor, information security director, network administrator, network engineer, and PC specialist. Chris has authored several print and online books on PC repair, CompTIA A+, CompTIA Security+, and Microsoft Windows. He has also served as technical editor and content contributor on numerous technical titles for several leading publishing companies. Chris holds numerous industry certifications, has been

recognized with many professional teaching awards, and has served as a state-level SkillsUSA competition judge.

About the Author Anthony Sequeira (CCIE No. 15626) began his IT career in 1994 with IBM in Tampa, Florida. He quickly formed his own computer consultancy, Computer Solutions, and then discovered his true passion: teaching and writing about networking technologies. Anthony lectured to massive audiences around the world while working for Mastering Computers. Anthony has never been happier in his career than he is now, as a senior technical instructor for Splunk. He is an avid tennis player, a private pilot, and a semi-professional poker player, and he loves anything at all to do with technology.

Way back in 1977, the International Organization for Standardization (ISO) developed a subcommittee to focus on the interoperability of multivendor communications systems. This is fancy language for getting network “thingies” to communicate with each other, even if different companies made those network “thingies.” What sprang from this subcommittee was the Open Systems Interconnection (OSI) reference model (also referred to as the OSI model or the OSI stack). Thanks to this model, you can talk about any networking technology and categorize that technology as residing at one or more of the seven layers of the model. This chapter defines those seven layers and offers examples of what you might find at each layer. It also contrasts the OSI model with another model —the TCP/IP stack, also known as the Department of Defense (DoD) model—that focuses on Internet Protocol (IP) communications.

If you were to look this or any other bookshelf in my home office, you would see that I have organized diverse types of books on different shelves. One shelf holds my collection of technical books, another shelf holds the books I wrote for Pearson and other publishers, another shelf holds books regarding self-improvement and finance. I have grouped similar books together on each shelf, just as the OSI model groups similar protocols and functions together in a layer. A common pitfall my readers meet when studying the OSI model is to try to neatly fit all the devices and protocols in their network into one of the OSI model’s seven layers. However, not every technology fits perfectly into these layers. In fact, some networks might not have any technologies running at one or more of these layers. This reminds me of my favorite statement about the OSI model. It comes from Rich Seifert’s book The Switch Book. In that book, Rich reminds us that the OSI model is a reference model, not a reverence model. That is, no cosmic law states that all technologies must cleanly plug into the model. So, as you discover the characteristics of the OSI model layers throughout this chapter, remember that these layers are like shelves for organizing similar protocols and functions, not immutable laws.

Note When first studying the OSI model, my students quickly realize that the model was created for the reasons described earlier. Later in their information technology (IT) careers, they realize the biggest value of the OSI model to them: to aid in troubleshooting network problems. Check out my video in the Additional Resources section at the end of this chapter, where I walk you through exactly how this is true!

The OSI Model As previously described, the OSI model consists of seven layers:

Layer 1: The physical layer Layer 2: The data link layer Layer 3: The network layer Layer 4: The transport layer Layer 5: The session layer Layer 6: The presentation layer Layer 7: The application layer Graphically, we depict these layers with Layer 1 at the bottom of the stack, as shown in Figure 1-2.

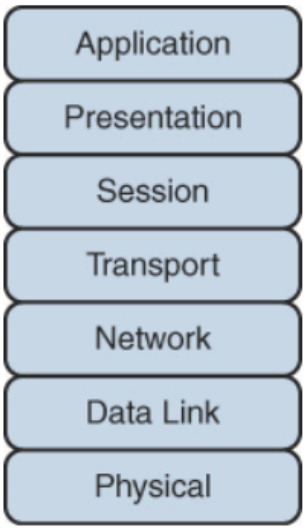
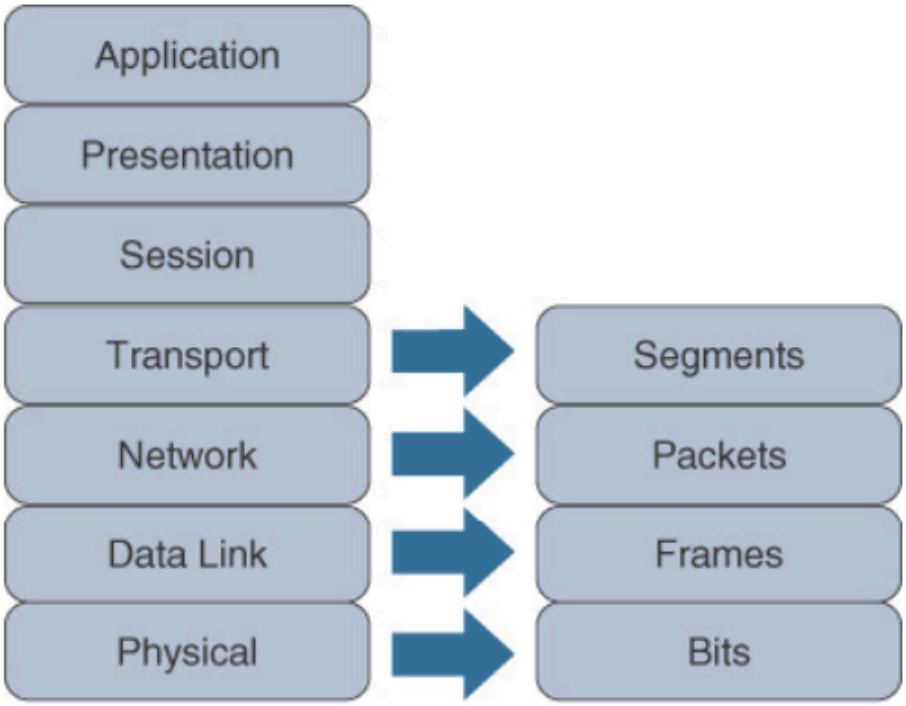


FIGURE 1-2 The OSI Stack

Note Various mnemonics are available to help memorize these layers in their proper order. A top-down (that is, starting at the top of the stack with Layer 7 and working your way down to Layer 1) memory aid is All People Seem To Need Data Processing. Another common technique is Please Do Not Throw Sausage Pizza Away, which begins at Layer 1 and works up to Layer 7. At the physical layer, binary expressions (that is, a series of 1s and 0s) represent data. A binary expression is created using bits, where a bit is a single 1 or a single 0. At upper layers, however, bits are grouped together, into what is known as a protocol data unit (PDU) or a data service unit. Engineers tend to use the term packet generically to refer to these PDUs. However, PDUs might have an added name, depending on their OSI layer. Figure 1-3 illustrates these PDU names. A common memory aid for these PDUs is Some People Fear Birthdays, where the S in Some reminds us of the S in Segments. The P in People reminds us of the P in Packets, and the F in Fear reflects the F in Frames. Finally, the B in Birthdays reminds us of the B in Bits. (If you have never heard this memory aid before, I am not that surprised as I invented it!)



URE 1-3 PDU Names

Layer 1: The Physical Layer The concern of the physical layer, as shown in Figure 1-4, is the transmission of bits on the network along with the physical and electrical characteristics of the network.

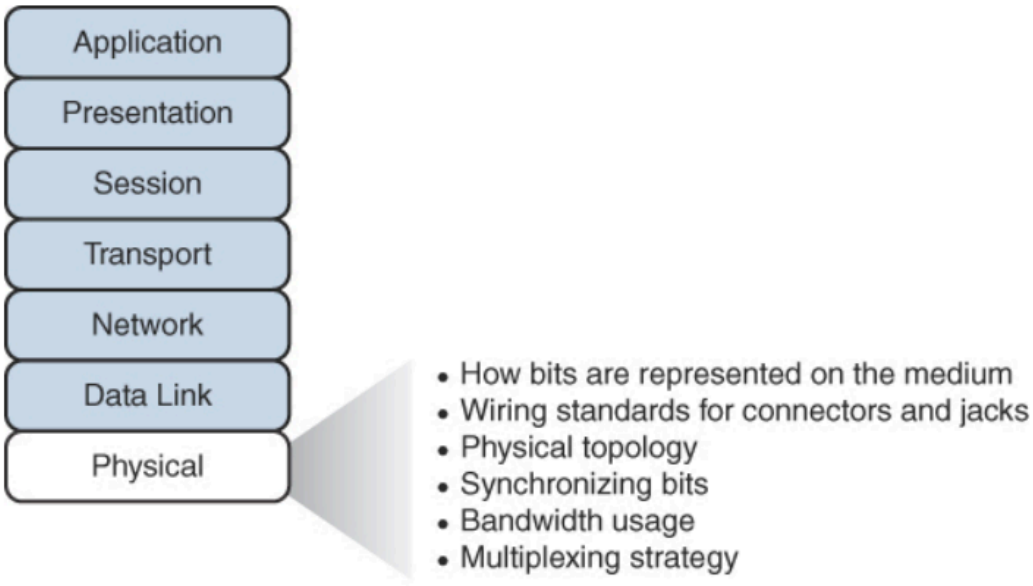


FIGURE 1-4 Layer 1: The Physical Layer

The physical layer defines the following: How to represent bits on the medium: Data on a computer network is represented as a binary expression. Chapter 4, “IP Addressing,” discusses binary in much more detail. Electrical voltage (on copper wiring) or light (carried via fiber-optic cabling) can represent these 1s and 0s. For example, the presence or absence of voltage on a wire portrays a binary 1 or a binary 0, respectively, as illustrated in Figure 1-5. Similarly, the presence or absence of light on a fiber-optic cable renders a 1 or 0 in binary. This type of approach is called current state modulation.

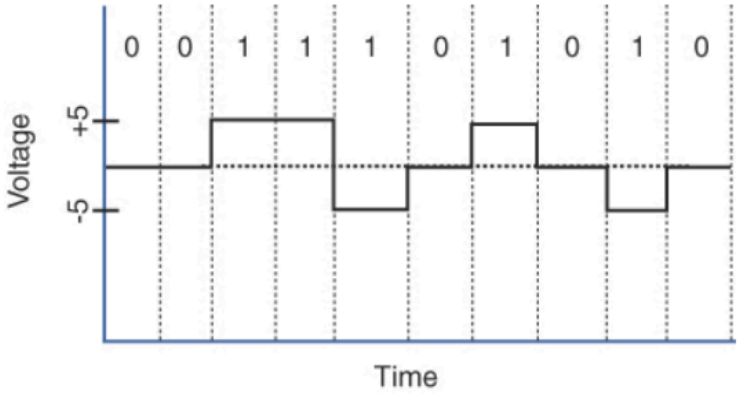


FIGURE 1-5 Current State Modulation

An alternative approach to portraying binary data is state transition modulation, as shown in Figure 1-6, where the transition between voltages or the presence of light shows a binary value.

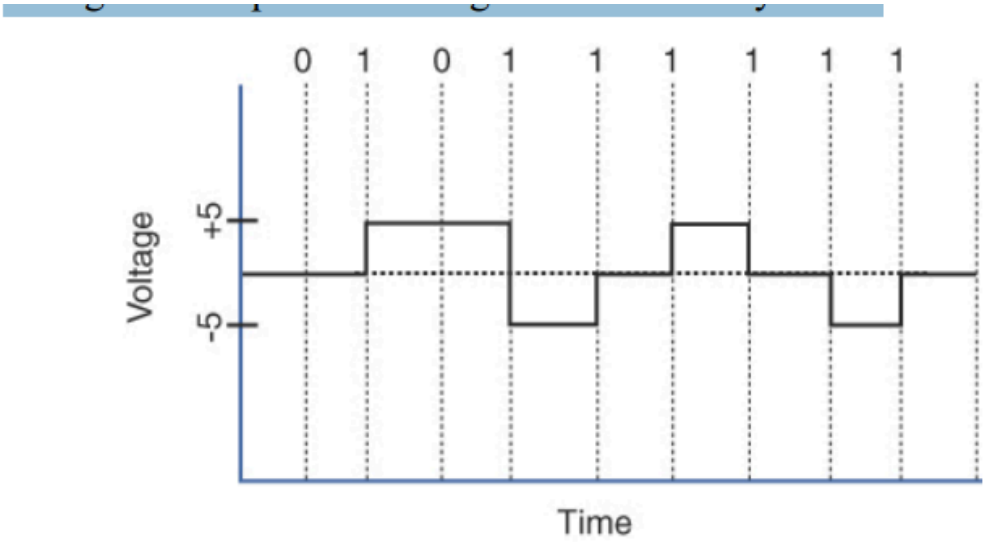


FIGURE 1-6 Transition Modulation

Note Other modulation types you might be familiar with from radio include amplitude modulation (AM) and frequency modulation (FM). AM uses a variation in a waveform's amplitude (that is, |||||||||||||||||||||||||||||||||||||||| signal strength) to portray the original signal. FM uses a variation in frequency to stand for the original signal.

Wiring standards for connectors and jacks: Chapter 3, "Network Media Types," describes several standards for network connectors. For example, the TIA/EIA-568-B standard describes how to wire an RJ-45 connector for use on a 100BASE-TX Ethernet network, as shown in Figure 1-7.

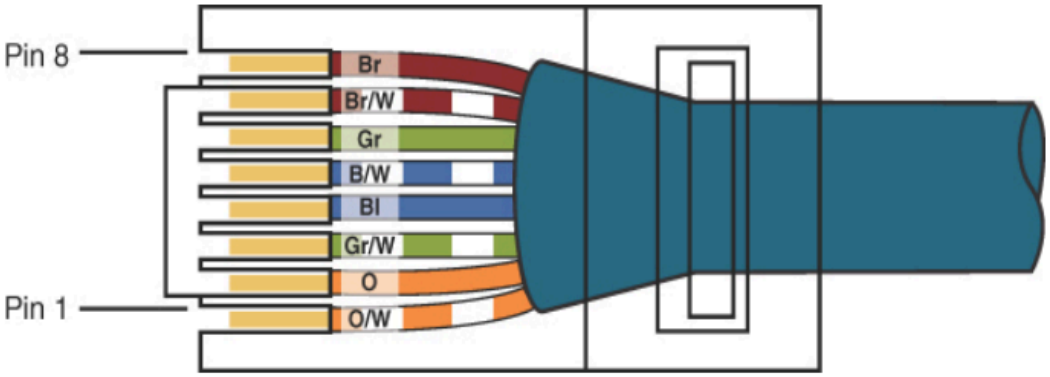


FIGURE 1-7 TIA/EIA-568-B Wiring Standard for an RJ-45 Connector

Physical topology: Layer 1 devices view a network as a physical topology (as opposed to a logical topology). Examples of a physical topology include bus, ring, and star topologies, as described in Chapter 2, “Network Topologies and Types.” Synchronizing bits: For two networked devices to successfully communicate at the physical layer, they must agree on when one bit stops and another bit starts. Specifically, the devices need a method to synchronize the bits. Two basic approaches to bit synchronization are asynchronous and synchronous synchronization: Asynchronous: With this approach, a sender states that it is about to start transmitting by sending a start bit to the receiver. When the receiver sees this, it starts its own internal clock to measure the next bits. After the sender transmits its data, it sends a stop bit to say that it has finished its transmission.

Synchronous: This approach synchronizes the internal clocks of the sender and the receiver to ensure that they agree on when bits begin and end. A common approach to make this synchronization happen is to use an external clock (for example, a clock provided by a service provider). The sender and receiver then reference this external clock. Bandwidth usage: The two fundamental approaches to bandwidth usage on a network are broadband and baseband: Broadband: Broadband technologies divide the bandwidth available on a medium (for example, copper or fiber-optic cabling) into different channels. A sender can then transmit different communication streams over the various channels. For example, consider frequency-division multiplexing (FDM) used by a cable modem. Specifically, a cable modem uses certain ranges of frequencies on the cable coming into your home from the local cable company to carry incoming data, another range of frequencies for outgoing data, and several other frequency ranges for various TV stations.

Baseband: Baseband technologies use all the available frequencies on a medium to send data. Ethernet is an example of a networking technology that uses baseband. Multiplexing strategy: Multiplexing allows multiple communications sessions to share the same physical medium. Cable TV, as previously mentioned, allows you to receive multiple channels over a single physical medium (for example, a coaxial cable plugged into the back of your television). Here are some of the most common approaches to multiplexing: Time-division multiplexing (TDM):

TDM supports different communication sessions (for example, different telephone conversations in a telephony network) on the same physical medium by causing the sessions to take turns. For a brief period, defined as a time slot, data from the first session is sent, followed by data from the second session. This continues until all sessions have had a turn, and the process repeats.

Statistical time-division multiplexing (StatTDM): A downside to TDM is that each communication session receives its own time slot, even if one of the sessions does not have any data to send at the moment. To make more efficient use of available bandwidth, StatTDM dynamically assigns time slots to communications sessions on an as-needed basis. Frequency-division multiplexing (FDM): FDM divides a medium's frequency range into channels, and different communication sessions send their data over different channels. As previously described, this approach to bandwidth usage is called broadband. Orthogonal frequency-division multiplexing (OFDM): OFDM encodes digital data onto multiple carrier frequencies. OFDM is very popular today and is used in wideband digital communication. This makes OFDM useful in applications such as digital television and audio broadcasting, DSL Internet access, wireless networks, powerline networks, and 4G/5G mobile communications. Examples of devices defined by physical layer standards include hubs, wireless access points, and network cabling. Note Hubs are not used in modern computer networks. So why are we even bothering to mention them? Well, they really did help give rise to our modern switches. A hub interconnects PCs in a LAN; it is considered a physical layer device because it takes bits coming in on one port and retransmits those bits out all other ports. At no point does the hub interrogate any addressing information in the data as our modern switches do.

Layer 2: The Data Link Layer Technet24 |||||||||||||||||||||||||||||||||||||||||||| The data link layer is concerned with the following: Packaging data into frames and transmitting those frames on the network Ensuring that frames do not exceed the maximum transmission unit (MTU) of the physical media Performing error detection/correction Uniquely finding network devices with addresses Handling flow control

Note Network interfaces use the MTU to define the largest packet size the interface will forward. For example, a 1500-byte packet could not be forwarded via a router interface with an MTU of 1470 bytes. Data link layer processes, collectively referred to as data link control (DLC), are illustrated in Figure 1-8.

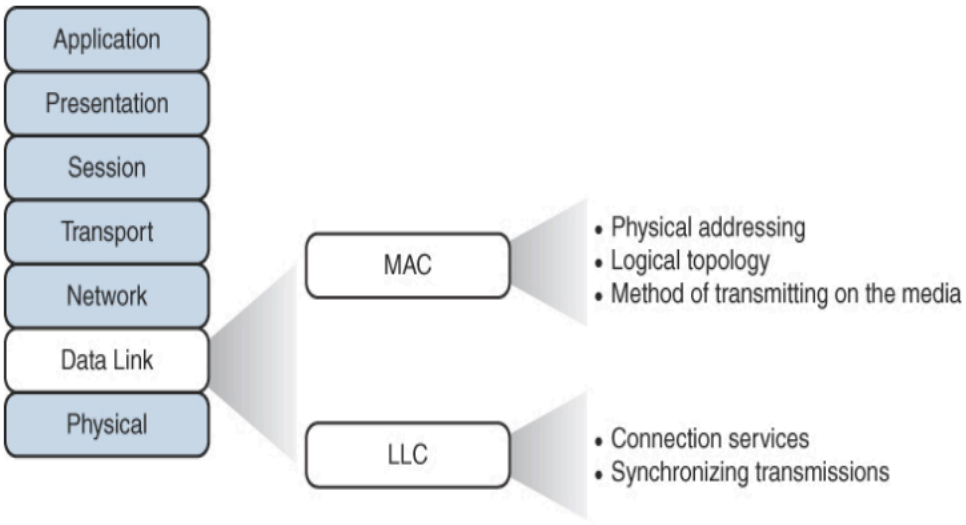


FIGURE 1-8 Layer 2: The Data Link Layer

In fact, the data link layer is distinct from the other layers in that it has two sublayers: MAC and LLC. Media Access Control Characteristics of the Media Access Control (MAC) sublayer of the data link layer include the following: Physical addressing: A common example of a Layer 2 address is a MAC address, which is a 48-bit address assigned to a device's network interface card (NIC). MAC addresses are written in hexadecimal notation (for example, 58:55:ca:eb:27:83). The first 24 bits of the 48- bit address are the vendor code. The IEEE Registration Authority assigns a manufacturer one or more unique vendor codes. You can use the list of vendor codes at https://standards.ieee.org/develop/regauth/oui/oui.txt to identify the manufacturer of a networking device, based on the first half of the device's MAC address. The last 24 bits of a MAC address are assigned by the manufacturer, and they act as a serial number for the device. No two MAC addresses in the world should have the same value.

Logical topology: Layer 2 devices view a network as a logical topology. Examples of logical topologies include bus and ring topologies, as described in Chapter 2. Method of transmitting on the media: With several devices connected to a network, there needs to be some strategy for deciding when a device sends on the media. Otherwise, multiple devices might send at the same time and interfere with one another's transmissions. Logical Link Control Characteristics of the Logical Link Control (LLC) sublayer of the data link layer include the following:

Connection services: When a device on a network receives a message from another device on the network, that recipient device can give feedback to the sender in the form of an acknowledgment message. The two main functions provided by these acknowledgment messages are as follows: Flow control: Limits the amount of data a sender can send at one time; this prevents the sender from overwhelming the receiver with too much information. Error control: Allows the recipient of data to let the sender know whether the expected data frame was

not received or whether it was received but is corrupted. The recipient figures out whether the data frame is corrupt by mathematically calculating a checksum of the data received. If the calculated checksum does not match the checksum received with the data frame, the recipient of the data draws the conclusion that the data frame is corrupted and can then notify the sender via an acknowledgment message. Synchronizing transmissions: Senders and receivers of data frames need to coordinate when a data frame is being transmitted and should be received. The three methods of performing this synchronization are detailed here:

Isochronous: With isochronous transmission, network devices look to a common device in the network as a clock source, which |||||||||||||||||||||||||||||||||||||||||| creates fixed-length time slots. Network devices can determine how much free space, if any, is available within a time slot and then insert data into an available time slot. A time slot can accommodate more than one data frame. Isochronous transmission does not need to provide clocking at the beginning of a data string (as does synchronous transmission) or for every data frame (as does asynchronous transmission). As a result, isochronous transmission uses little overhead compared to asynchronous or synchronous transmission methods.

Asynchronous: With asynchronous transmission, network devices reference their own internal clocks, and network devices do not need to synchronize their clocks. Instead, the sender places a start bit at the beginning of each data frame and a stop bit at the end of each data frame. These start and stop bits tell the receiver when to monitor the medium for the presence of bits. An additional bit, called the parity bit, might also be added to the end of each byte in a frame to detect an error in the frame. For example, if even parity error detection (as opposed to odd parity error detection) is used, the parity bit (with a value of either 0 or 1) would be added to the end of a byte, causing the total number of 1s in the data frame to be an even number. If the receiver of a byte is configured for even parity error detection and receives a byte where the total number of bits (including the parity bit) is even, the receiver can conclude that the byte was not corrupted during transmission. Note Using a parity bit to detect errors might not be effective if a byte has more than one error (that is, if more than one bit has been changed from its original value).

Synchronous: With synchronous transmission, two network devices that want to communicate between themselves must agree on a clocking method to show the beginning and ending of data frames. One approach to providing this clocking is to use a separate Technet24 |||||||||||||||||||||||||||||||||||||||||| communications channel over which a clock signal is sent. Another approach relies on specific bit combinations or control characters to indicate the beginning of a frame or a byte of data. Like asynchronous transmissions, synchronous transmissions can perform error detection. However, rather than using parity bits, synchronous communication runs a mathematical algorithm on the data to create a cyclic redundancy check (CRC). If the sender and the receiver calculate the same CRC value for the same chunk of data, the receiver can conclude that the data was not corrupted during transmission.

Examples of devices defined by data link layer standards include switches, bridges, and NICs. Note NICs are not entirely defined at the data link layer because they are partially based on

physical layer standards, such as a NIC's network connector. Layer 3: The Network Layer The network layer, as shown in Figure 1-9, is primarily concerned with forwarding data based on logical addresses.

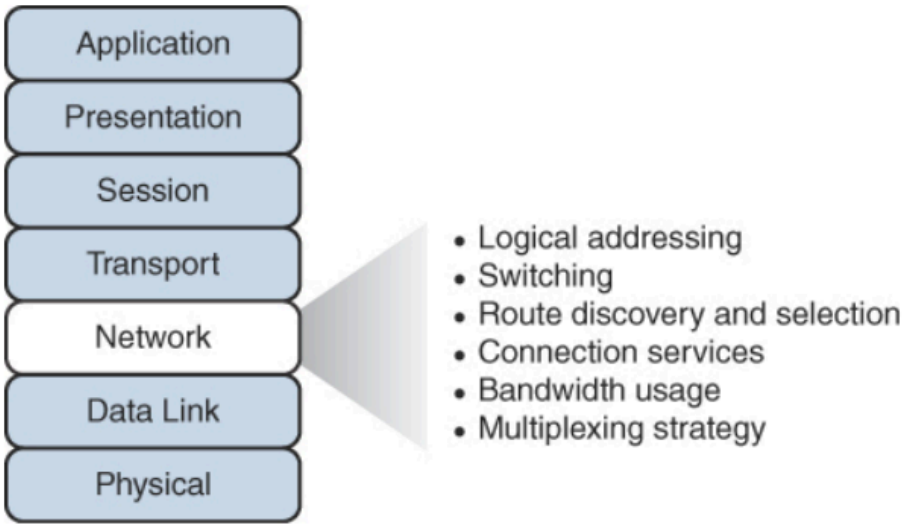


FIGURE 1-9 Layer 3: The Network Layer

Although many network administrators think of routing and IP addressing when they hear about the network layer, this layer is actually responsible for a variety of tasks: Logical addressing: Whereas the data link layer uses physical addresses to make forwarding decisions, the network layer uses logical addressing to make forwarding decisions. A variety of routed protocols (for example, AppleTalk and IPX) have their own logical addressing schemes, but by far the most widely deployed routed protocol is Internet Protocol (IP). Chapter 4 discusses IP addressing in detail. Switching: Engineers often associate the term switching with Layer 2 technologies; however, the concept of switching also exists at Layer 3. Switching, at its essence, is making decisions about how data should be forwarded. At Layer 3, three common switching techniques exist: Packet switching: With packet switching, a data stream is divided into packets. Each packet has a Layer 3 header that includes source and destination Layer 3 addresses. Another term for packet switching is routing, which is discussed in more detail in Chapter 10, "Routing Technologies and Bandwidth Management."

Circuit switching: Circuit switching dynamically brings up a dedicated communication link between two parties for those parties to communicate.

As a simple example of circuit switching, think of making a phone call from your home to your business. In fact, let's go "old school" and pretend you have a traditional landline servicing your phone, the telephone company's switching equipment interconnects your home phone with the

phone system of the business you are calling. This interconnection (that is, *circuit*) exists only for the duration of the phone call.

Message switching: Unlike packet switching and circuit switching technologies, message switching is usually not well suited for realtime applications because of the delay involved. Specifically, with message switching, a data stream is divided into messages. Each message is tagged with a destination address, and the messages travel from one network device to another network device on the way to their destination. Because these devices might briefly store the messages before forwarding them, a network using message switching is sometimes called a store-and-forward network. Metaphorically, you could visualize message switching like routing an email message, where the email message might be briefly stored on an email server before being forwarded to the recipient.

Route discovery and selection: Because Layer 3 devices make forwarding decisions based on logical network addresses, a Layer 3 device might need to know how to reach various network addresses. For example, a common Layer 3 device is a router. A router can maintain a routing table indicating how to forward a packet based on the packet's destination network address. A router can have its routing table populated via manual configuration (that is, by entering static routes), via a dynamic routing protocol (for example, OSPF or EIGRP), or simply by being directly connected to certain networks.

Connection services: Just as the data link layer offers connection services for flow control and error control, connection services also exist at the network layer. Connection services at the network layer can improve the communication reliability if the data link's LLC sublayer is not performing connection services. The following functions are performed by connection services at the network layer: Flow control (also known as congestion control): Helps prevent a sender from sending data more rapidly than the receiver is capable of receiving it. Packet reordering: Allows packets to be placed in the proper sequence as they are sent to the receiver. This might be necessary because some networks support load balancing, where multiple links are used to send packets between two devices. Because multiple links exist, packets might arrive out of order. Examples of devices found at the network layer include routers and multilayer switches. The most common Layer 3 protocol in use, and the protocol on which the Internet is based, is IPv4. However, IPv6 is beginning to be more common on networks today.

Layer 4: The Transport Layer The transport layer, as shown in Figure 1-10, acts as a dividing line between the upper layers and lower layers of the OSI model. Specifically, messages are taken from upper layers (Layers 5–7) and are encapsulated into segments for transmission to the lower layers (Layers 1–3). Similarly, data streams coming from lower layers are decapsulated and sent to Layer 5 (the session layer), or some other upper layer, depending on the protocol.

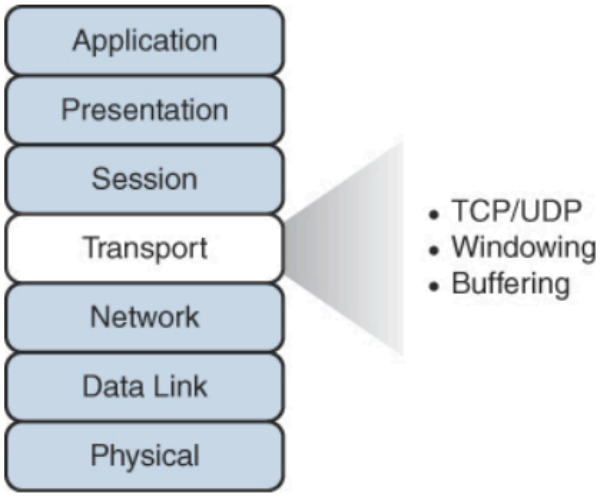


FIGURE 1-10 Layer 4: The Transport Layer

Two common transport layer protocols are TCP and UDP: Transmission Control Protocol (TCP): TCP is a connection-oriented transport protocol. Connection-oriented transport protocols offer reliable transport, in that if a segment is dropped, the sender can detect the drop and retransmit the dropped segment. Specifically, a receiver acknowledges segments that it receives. Based on those acknowledgments, a sender can decide which segments were successfully received and which segments need to be transmitted again. User Datagram Protocol (UDP): UDP is a connectionless transport protocol. Connectionless transport protocols offer unreliable transport, in that if a segment is dropped, the sender is unaware of the drop, and no retransmission occurs. Just as Layer 2 and Layer 3 offer flow control services, flow control services also exist at Layer 4. Two common flow control approaches at Layer 4 are windowing and buffering:

- Windowing: TCP communication uses windowing, in that one or more segments are sent at one time, and a receiver can attest to the receipt of all the segments in a window with a single acknowledgment. In some cases, as illustrated in Figure 1-11, TCP uses a sliding window, where the window size begins with one segment. If there is a successful acknowledgement of that one segment (that is, the receiver sends an acknowledgement asking for the next segment), the window size doubles to two segments. Upon successful receipt of those two segments, the next window holds four segments. This exponential increase in window size continues until the receiver does not acknowledge successful receipt of all segments within a certain amount of time – also known as the *round-trip time*, which is sometimes called *real transfer time* – or until a configured maximum window size is reached.

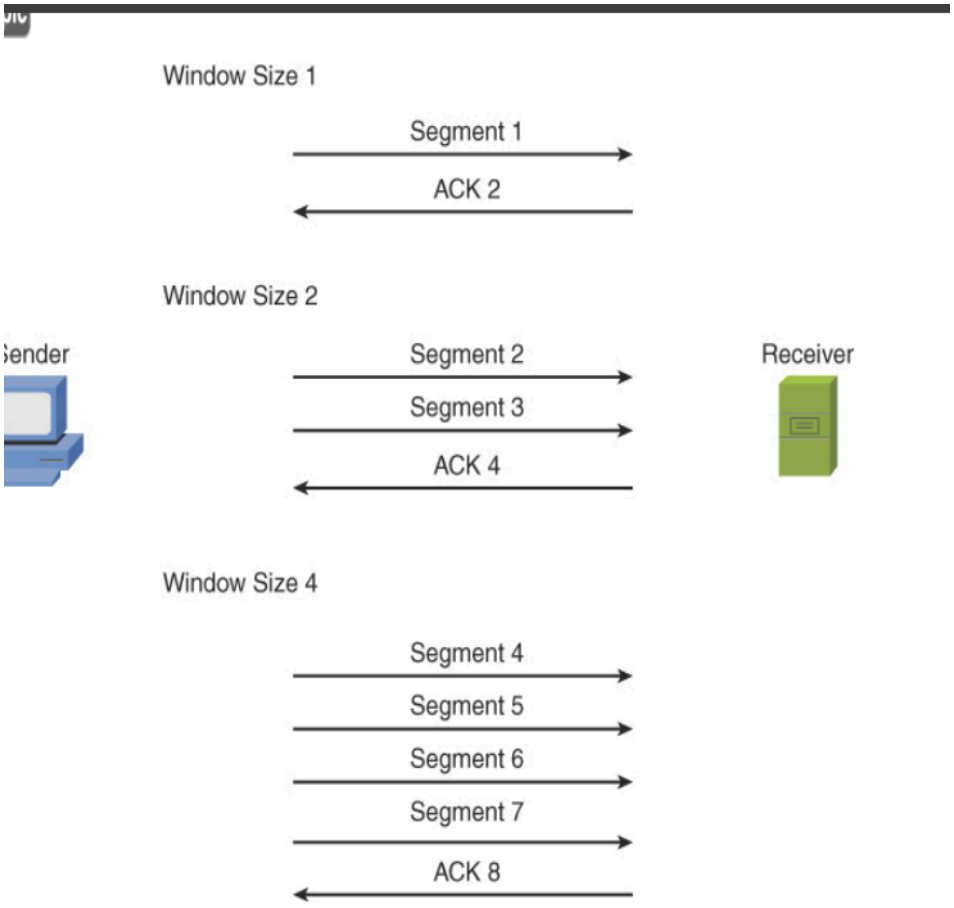


FIGURE 1-11 TCP Sliding Window Buffering: With buffering, a device (for example, a router) uses a chunk of memory (sometimes called a buffer or a queue) to store segments if bandwidth is not available to send those segments. A queue has finite capacity, however, and can overflow (that is, drop segments) in the event of sustained network congestion. In addition to TCP and UDP, Internet Control Message Protocol (ICMP) is another transport layer protocol you are likely to meet. ICMP is used by utilities such as ping and traceroute, which are discussed in Chapter 23, “Network Software Tools and Commands.” Layer 5: The Session Layer The session layer, as shown in Figure 1-12, is responsible for setting up, maintaining, and tearing down sessions. You can think of a session as a conversation that needs to be treated separately from other sessions to avoid the intermingling of data from different conversations.

the intermingling of data from different conversations.

Key Topic

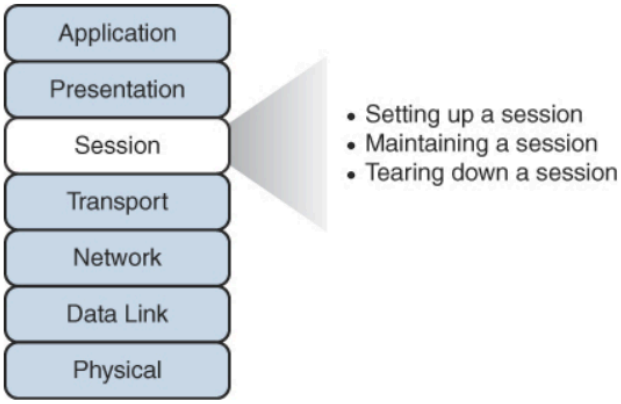


FIGURE 1-12 Layer 5: The Session Layer

Here is a detailed look at the functions of the session layer: Setting up a session: Examples of the procedures involved in setting up a session include the following: Checking user credentials (for example, username and password) Assigning numbers to a session's communication flows to uniquely find each one Negotiating services needed during the session Negotiating which device begins sending data Maintaining a session: Examples of the procedures involved in supporting a session include the following: Transferring data Reestablishing a disconnected session Acknowledging receipt of data Tearing down a session: A session can be disconnected based on agreement of the devices in the session. Alternatively, a session might be torn down because one party disconnects (either intentionally or because of an error condition). If one party disconnects, the other party can detect a loss of communication with that party and tear down its side of the session.

Session Initiation Protocol (SIP) is an example of a session layer protocol, which can help set up, support, and tear down a voice or video connection. Keep in mind, however, that not every network application neatly maps directly to all seven layers of the OSI model. The session layer is one of those layers where it might not be possible to name what protocol in each scenario is running in it. Network Basic Input/Output System (NetBIOS) is one example of a session layer protocol. Note NetBIOS is an application programming interface (API) developed in the early 1980s to enable computer-to-computer communication on a small LAN (specifically, PC-Network, which was IBM's LAN technology at the time). Later, IBM needed to support computer-tocomputer communication over larger Token Ring networks. As a result, IBM enhanced the scalability and features of NetBIOS with a NetBIOS emulator named NetBIOS Extended User Interface (NetBEUI). Layer 6: The Presentation Layer The presentation layer, as shown in Figure 1-13, formats the data being exchanged and secures that data with encryption.

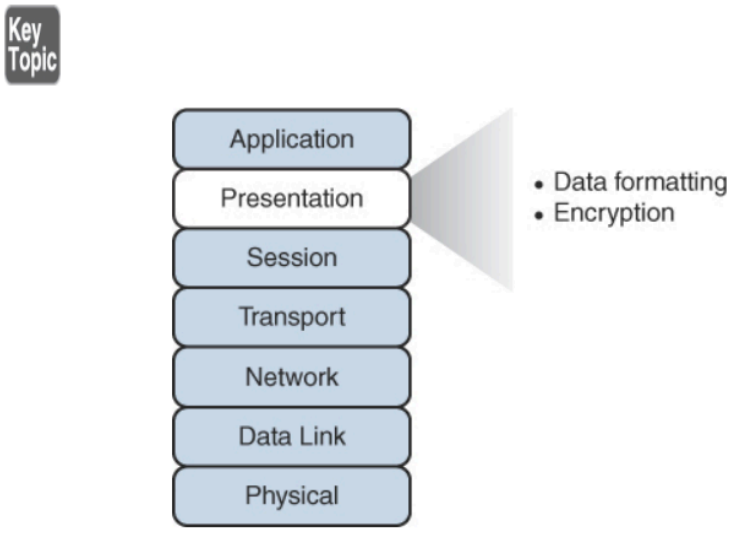


FIGURE 1-13 Layer 6: The Presentation Layer

FIGURE 1-13 Layer 6: The Presentation Layer The following list describes the functions involved in data formatting and encryption in more detail: Data formatting: As an example of how the presentation layer handles data formatting, consider how text is formatted. Some applications might format text using American Standard Code for Information Interchange (ASCII), while other applications might format text using Extended Binary Coded Decimal Interchange Code (EBCDIC). The presentation layer handles formatting the text (or other types of data, such as multimedia or graphics files) in a format that allows compatibility between the communicating devices.

Encryption: Imagine that you are sending sensitive information over a network (for example, your credit card number or bank password). If a malicious user were to intercept your transmission, they might be able to obtain this sensitive information. To add a layer of security for such transmissions, encryption can be used to scramble (encrypt) the data in such a way that if the data were intercepted, a third party would not be able to unscramble (decrypt) it. However, the intended recipient would be able to decrypt the transmission. Encryption is discussed in detail in Chapter 16, “Common Security Concepts.” Layer 7: The Application Layer The application layer, as shown in Figure 1-14, gives application services to a network. An important (and often-misunderstood) concept is that end user applications (such as Microsoft Word) live at the application layer. Instead, the application layer supports services used by end-user applications. For example, email is an application layer service that does exist at the application layer, whereas Microsoft Outlook (an example of an email client) is an end-user application that does not live at the application layer. Another function of the application layer is advertising available services.

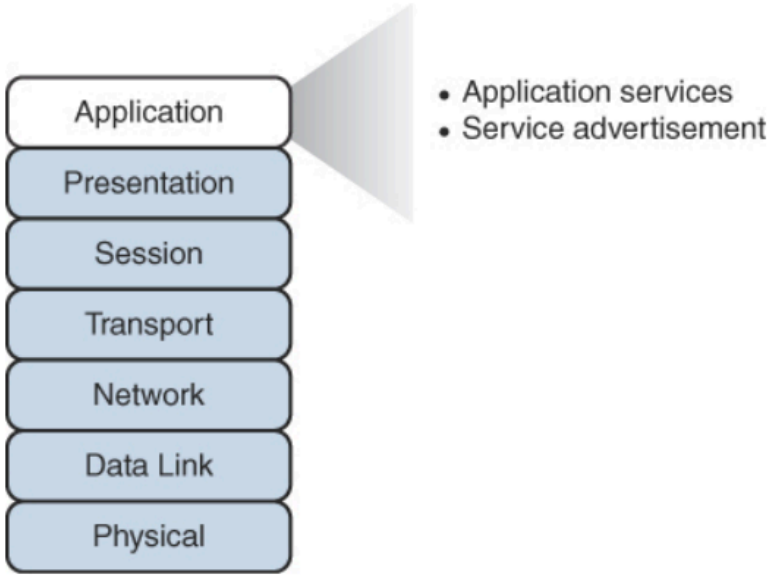


FIGURE 1-14 Layer 7: The Application Layer

The following list describes the functions of the application layer in more detail: Application services: Examples of the application services living at the application layer include file sharing and email. Service advertisement: Some applications' services (for example, some networked printers) periodically send out advertisements, making their availability known to other devices on the network. Other services, however, register themselves and their services with a centralized directory (for example, Microsoft Active Directory), which can be queried by other network devices seeking such services. Recall that even though the application layer is numbered as Layer 7, it is at the top of the OSI stack because its networking functions are closest to the end user.

The TCP/IP Stack The ISO developed the OSI reference model to be generic, in terms of what protocols and technologies could be categorized by the model. However, |||||||||||||||||||||||||||||||||||||||||||||||| most of the traffic on the Internet (and traffic on corporate networks) is based on the TCP/IP protocol suite. Therefore, a more relevant model for many network designers and administrators to reference is a model developed by the U.S. Department of Defense (DoD). This model is known as the DoD model, or the TCP/IP stack.

Note An older protocol known as Network Control Protocol (NCP) was similar to TCP/IP. NCP was used on ARPANET (the predecessor to the Internet), and it provided features like those offered by the TCP/IP suite of protocols on the Internet, although they were not as robust. Layers of the TCP/IP Stack The TCP/IP stack has only four defined layers, as opposed to the seven layers of the OSI model. Figure 1-15 contrasts these two models. The TCP/IP stack is composed of the following layers: Network interface: The TCP/IP stack's network interface layer

encompasses the technologies offered by Layers 1 and 2 (the physical and data link layers) of the OSI model.

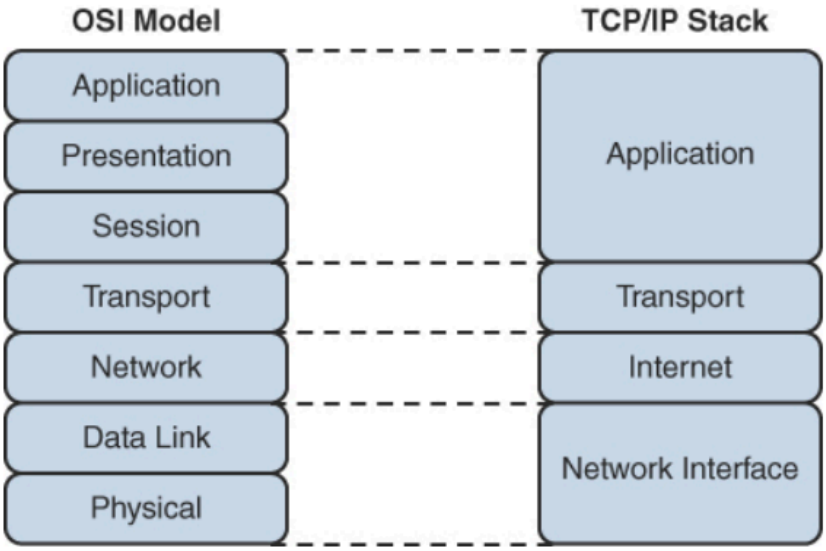


FIGURE 1-15 TCP/IP Stack

Some literature refers to the network interface layer as the *network access layer.*

Internet: The Internet layer of the TCP/IP stack maps to Layer 3 (the network layer) of the OSI model. Although multiple routed protocols live at the OSI model's network layer, the Internet layer of the TCP/IP stack focuses on IP as the protocol to be routed through a network. Figure 1-16 shows the format of an IP version 4 (IPv4) packet.

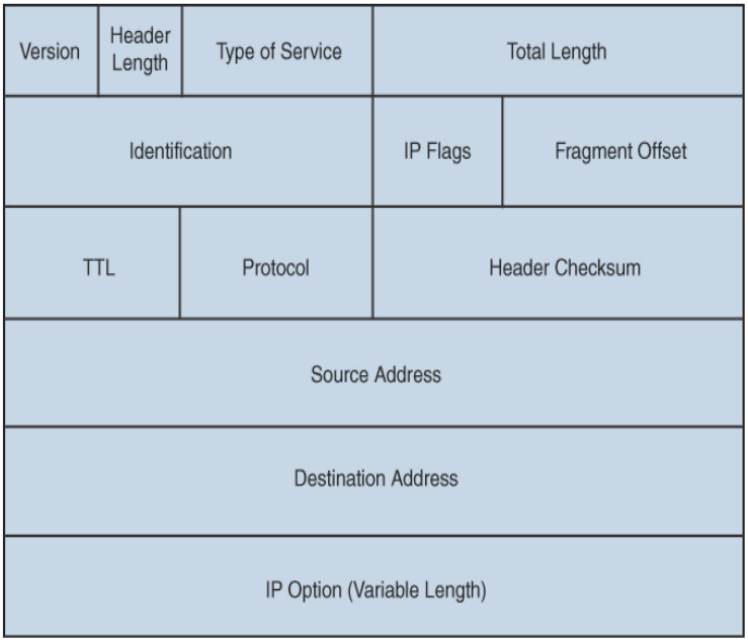


FIGURE 1-16 IPv4 Packet Format

Notice that there are fields in the IP packet header for both source and a destination IP addresses. The Protocol field shows the transport layer protocol from which the packet was sent or to which the packet should be sent. Also of note is the Time-to-Live (TTL) field. The value in this field is decremented by 1 every time this packet is routed from one IP network to another (that is, when it passes through a router). If the TTL value ever reaches 0, the packet is discarded from the network. This behavior helps prevent routing loops. As a common practice, the OSI layer numbers 1, 2, and 3 are still used when referring to physical, data link, and network layers of the TCP/IP stack, even though the TCP/IP stack does not explicitly separate the physical and data link layers.

Transport: The *transport layer* of the TCP/IP stack maps to Layer 4 (the transport layer) of the OSI model. The two primary protocols found at the TCP/IP stack's transport layer are TCP and UDP.

Figure 1-17 details the structure of a TCP segment. Notice the fields for source and destination ports. As described later in this chapter, these ports identify to which upper-layer protocol data should be forwarded or from which upper-layer protocol the data is being sent.

Key Topic

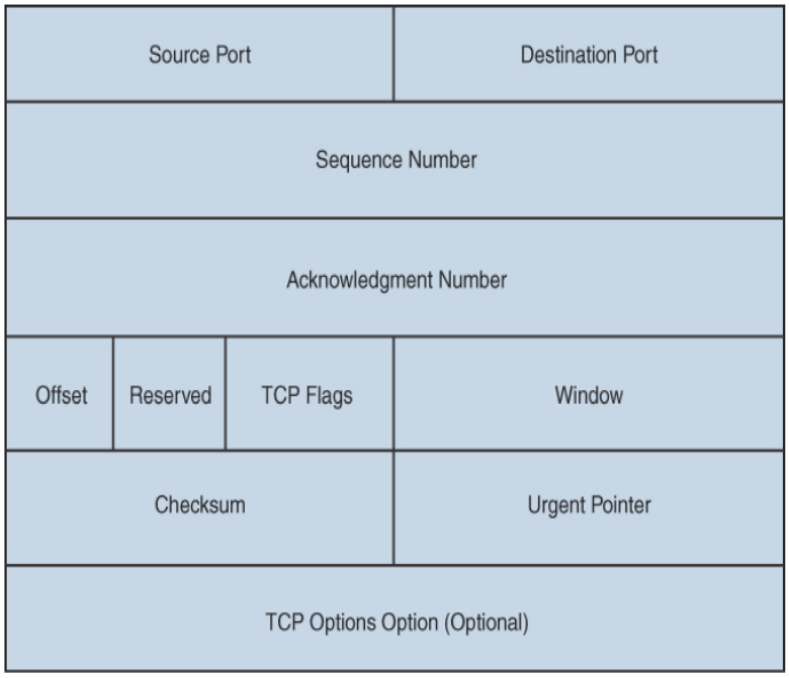


FIGURE 1-17 TCP Segment Format

Also notice the field for window size. The value in this field determines how many bytes a device can receive before expecting an |||||||||||||||||||||||||||||||||||||||| acknowledgment. As previously described, this feature offers flow control. The header of a TCP segment also contains sequence numbers for segments. With sequence numbering, if segments arrive out of order, the recipient can put them back in the proper order based on the sequence numbers. The acknowledgement number in the header shows the next sequence number the receiver expects to receive. This is a way for the receiver to let the sender know that all segments up to and including to that point have been received. Due to the sequencing and acknowledgements, TCP is considered to be a *connection-oriented* transport layer protocol.

Note: You might have noticed that both the *IP header* and *TCP header* make use of a Flags field. Including this field is a very common technique in networking to permit the data unit to permit the data unit to convey specific settings. For example, the IP header uses the IP Flags (3 bits) to help manage (or prevent) fragmentation. *TCP flags* are used to indicate a particular connection state or provide additional information. They are often used for troubleshooting purposes or to control how a particular connection is handled.

Figure 1-18 presents the structure of a UDP segment. UDP is a connectionless, unreliable protocol. UDP lacks the sequence number, window size, and acknowledgement numbering present in the header of a TCP segment. The UDP segment's header simply contains source

and destination port numbers, a UDP checksum (which is an optional field to detect transmission errors), and the segment length (measured in bytes).

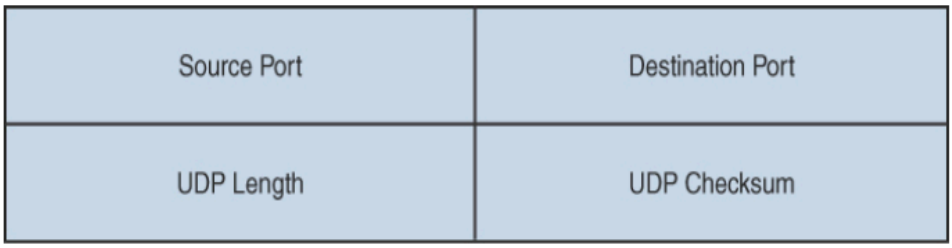


FIGURE 1-18 UDP Segment Format

Because a UDP header is much smaller than a TCP header, UDP is a good candidate for the transport layer protocol for applications that need to maximize bandwidth and do not require acknowledgments (for example, audio or video streams). Application: The biggest difference between the TCP/IP stack and the OSI model is at the TCP/IP stack's application layer. This layer addresses concepts described by Layers 5, 6, and 7 (the session, presentation, and application layers) of the OSI model. With the reduced complexity of a four-layer model like the TCP/IP stack, network designers and administrators can more easily categorize a given networking technology into a specific layer. For example, although SIP was shown earlier as a session layer protocol within the OSI model, you would have to know more about the behavior of SIP to properly categorize it in that model. However, with the TCP/IP stack, you could quickly figure out that SIP is a higher-level protocol that gets encapsulated inside TCP, and you could thus classify SIP in the application layer of the TCP/IP stack. Common Application Protocols in the TCP/IP Stack Application layer protocols in the TCP/IP stack are identifiable by unique port numbers. For example, when you enter a web address in an Internet browser, you are (by default) communicating with that remote web address using TCP port 80. Specifically, Hypertext Transfer Protocol (HTTP), which is the protocol used by web servers, uses TCP port 80. Therefore, the data you send to that remote web server has a destination port number of 80. That data is encapsulated into a TCP segment at the transport layer. That segment is further encapsulated into a packet at the Internet layer and sent out on the network using an underlying network interface layer technology such as Ethernet. Note Thanks to awareness of network security today, you do not see HTTP (port 80) actually being used on the Internet very much anymore. It has been replaced with a secured version, HTTPS, which uses TCP port 443 in its operation. Consider the example illustrated in Figure 1-19. When you send traffic to the remote website, the packet you send out to the network needs not only the destination IP address (172.16.1.2 in this example) of the web server and the destination port number for HTTP (that is, 80) but also the source IP address of your computer (10.1.1.1 in this example). Because your computer is not acting as a web server, its port is not 80. Instead, your computer selects a

source port number greater than 1023. In this example, let's imagine that the client PC selects the source port 1248.

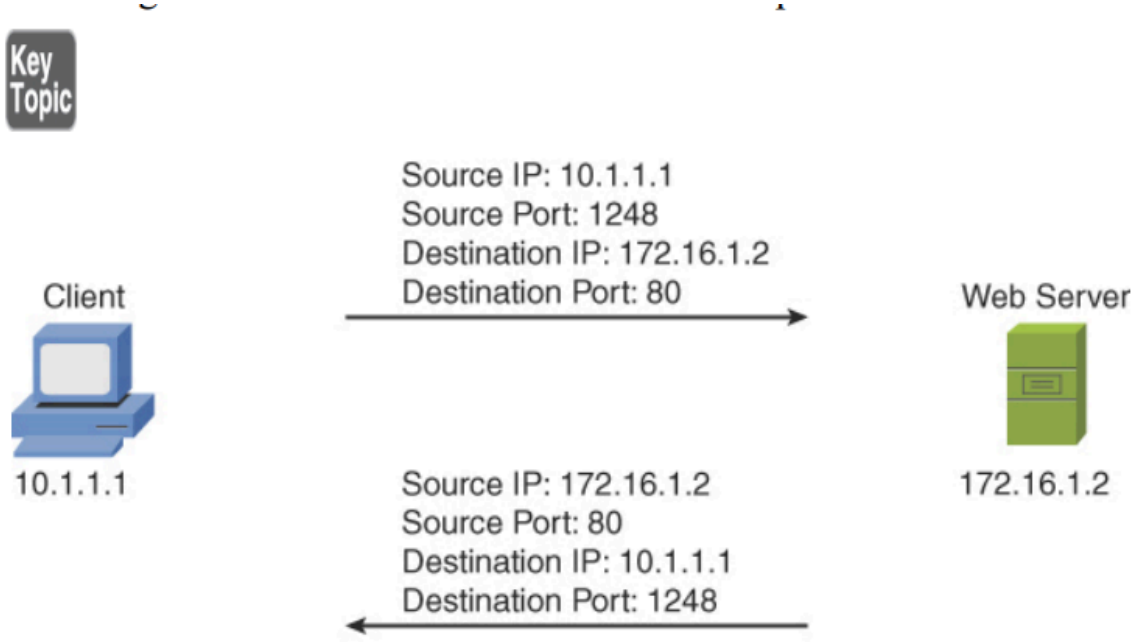


FIGURE 1-19 Example: Port Numbers and IP Addresses

Notice that when the web server sends content back, the IP addresses and port numbers have now switched, with the web server as the source and your PC as the destination. With both source and destination port numbers, along with source and destination IP addresses, two-way communication becomes possible. Note Ports numbered 1023 and below are called well-known ports, and ports numbered above 1023 are called ephemeral ports. The maximum value of a port is 65,535. Well-known port number assignments are available at https://www.iana.org/assignments/portnumbers.

Real World Case Study

Bob, a manager of the networking team at Acme, Inc. is paying extra attention to specific words he uses as he talks to his team in preparation for implementation of the network. When referring to transport protocols such as the connection-oriented TCP and the connectionless UDP, the word Bob uses to describe those protocol data units is *segment*. In discussing the many applications that the company will be using over its network, Bob notes that many of these applications will be using TCP at the transport layer. This includes HTTPS for secure web traffic and Simple Mail Transfer Protocol (SMTP) and Internet Message Access Protocol (IMAP) for email services. The company will use the Secure Shell (SSH) protocol, which also uses TCP at the transport layer. This includes HTTPS for secure web traffic and Simple Mail Transfer Protocol (SMTP) and Internet Message Access Protocol (IMAP) for email services. The company will use the Secure Shell (SSH) protocol, which also uses TCP at the transport layer,

as a secure method to remotely connect to and manage its network devices. A common connectionless UDP protocol is Domain Name System (DNS), which will be used thousands of times a day to translate a friendly name like http://www.pearson.com to an IP address that is reachable over the network. Another protocol based on UDP that will be used often is Dynamic Host Configuration Protocol (DHCP), which assigns client computers on the network an IP address that is required for sending and receiving Layer 3 packets.

For the traffic on the LAN, the Ethernet cables and electronic signals being sent as bits going over those cables represent Layer 1 from an OSI perspective. On the LAN, they will be using Ethernet technology, and as a result, the Layer 2 frames that are sent on the LAN will be encapsulated and sent as Ethernet Layer 2 frames. For datagrams being sent across the serial WAN connections provided by the service provider, it is likely that either Point-to-Point Protocol (PPP) or High-Level Data Link Control (HDLC) encapsulation will be used for the Layer 2 frames. On both the LAN and the WAN, at Layer 3 (the network layer), IPv4 will be used for host addressing and defining networks. The same Layer 1, Layer 2, and Layer 3 infrastructure is also capable of transporting IPv6, if desired. Inside the Layer 3 IP headers, each packet contains the source and destination address, in addition to the information to tell the receiving network device about which Layer 4 transport protocol is encapsulated or carried inside the Layer 3 packet. When a network device receives the packet and opens it up to look at the contents, this process is called decapsulation. As the recipient decapsulates and looks at the Layer 4 information, it identifies the application layer protocol or service being used. A segment going to a web server is likely to have a TCP destination port of 80 or 443, depending on whether encryption is being used for a secure connection. A DNS request uses a UDP destination port of 53.

Summary

For the traffic on the LAN, the Ethernet cables and electronic signals being sent as bits going over those cables represent Layer 1 from an OSI perspective. On the LAN, they will be using Ethernet technology, and as a result, the Layer 2 frames that are sent on the LAN will be encapsulated and sent as Ethernet Layer 2 frames. For datagrams being sent across the serial WAN connections provided by the service provider, it is likely that either Point-to-Point Protocol (PPP) or High-Level Data Link Control (HDLC) encapsulation will be used for the Layer 2 frames. On both the LAN and the WAN, at Layer 3 (the network layer), IPv4 will be used for host addressing and defining networks. The same Layer 1, Layer 2, and Layer 3 infrastructure is also capable of transporting IPv6, if desired. Inside the Layer 3 IP headers, each packet contains the source and destination address, in addition to the information to tell the receiving network device about which Layer 4 transport protocol is encapsulated or carried inside the Layer 3 packet. When a network device receives the packet and opens it up to look at the contents, this process is called decapsulation. As the recipient decapsulates and looks at the Layer 4 information, it identifies the application layer protocol or service being used. A segment going to a web server is likely to have a TCP destination port of 80 or 443, depending on whether encryption is being used for a secure connection. A DNS request uses a UDP destination port of 53. interface, Internet, transport, and application. These layers are compared with the seven

layers of the OSI model. Data encapsulation and decapsulation within the OSI model context are covered. This chapter discusses how port numbers are used to associate data at the transport layer with a proper application layer protocol.

Table 1-1 Key Topics for Chapter 1 Key Topic Element Description Page Number List Layers of the OSI model 6 Figure 1-3 Protocol data unit (PDU) names 7 Figure 1-4 Layer 1: The physical layer 8 Figure 1-8 Layer 2: The data link layer 12 Figure 1-9 Layer 3: The network layer 15 Figure 1-10 Layer 4: The transport layer 17 Figure 1-11 TCP sliding window 18 Figure 1-12 Layer 5: The session layer 19 Figure 1-13 Layer 6: The presentation layer 20

Figure 1-14 Layer 7: The application layer 21 Figure 1-15 TCP/IP stack 23 Figure 1-16 IPv4 packet format 23 Figure 1-17 TCP segment format 24 Figure 1-18 UDP segment format 25 Figure 1-19 Example: Port numbers and IP addresses 26 Table 1-1 Application layer protocols/applications 29

Define Key Terms

Define Key Terms Define the following key terms from this chapter and check your answers in the Glossary: Open Systems Interconnection (OSI) reference model protocol data unit (PDU) IP header TCP header UDP header TCP flags payload maximum transmission unit (MTU) current state modulation state transition modulation cyclic redundancy check (CRC) physical layer data link layer network layer transport layer (OSI model) session layer presentation layer application layer (OSI model)

Network interface layer
Internet layer
Transport layer (TCP/IP) stack
Application layer (TCP/IP) stack
Time-division multiplexing (TDM)
Transmission Control Protocol (TCP)

Additional resources

Troubleshooting with the OSI model: https://youtu.be/kdFOCIeUkVE
The OSI Model Challenge: https://ajsnetworking.com/osiquiz1

Review Questions The answers to these review questions appear in Appendix A, “Answers to Review Questions.” 1. Which layer of the OSI reference model contains the MAC and LLC sublayers? a. Network layer b. Transport layer c. Physical layer d. Data link layer

2. Which approach to bandwidth usage consumes all the available frequencies on a medium to transmit data? a. Broadband b. Baseband c. Time-division multiplexing d. Simplex 3. Windowing is provided at what layer of the OSI reference model? a. Data link layer b. Network layer c.

Transport layer d. Physical layer 4. IP addresses reside at which layer of the OSI reference model? a. Network layer b. Session layer c. Data link layer d. Transport layer

5. Which of the following is a connectionless transport layer protocol? a. IP b. TCP c. UDP d. SIP 6. What setting ultimately controls the size of packets that are moving through the modern network? a. TTL b. MTU c. SSH

d. CSMA/CD 7. What is the range of well-known TCP and UDP ports? a. Below 2048 b. Below 1024 c. 16,384–32,768 d. Above 8192 8. What port number is used by HTTPS? a. 80 b. 443 c. 69 d. 23 9. What value is decremented by one for each router hop on the network? a. Count b. Type c. TTL d. Dead timer 10. Windowing is a technology that applies to which transport layer protocol? a. UDP b. FTP c. ICMP d. TCP 11. What happens to data as it moves from the upper layers to the lower layers of the OSI model on a host system? a. The data moves from the physical layer to the application layer. b. The data is encapsulated with a header at the beginning and a trailer at the end. c. The header and trailer are stripped off through decapsulation. d. The data is sent in groups of segments that require two acknowledgments. 12. Which layer of the OSI reference model is responsible for ensuring that frames do not exceed the maximum transmission unit (MTU) of the physical media? a. Network layer b. Transport layer c. Physical layer d. Data link layer

Chapter 2: Network Topologies and Types

This chapter covers the following topics related to Objective 1.2 (Explain the characteristics of network topologies and network types) of the CompTIA Network+ N10-008 certification exam:

- Mesh
- Star/Hub and spoke
- Bus
- Ring
- Hybrid
- Network types and characteristics
    - Peer-to-peer
    - Client-server
    - Local area network (LAN)
    - Metropolitan area network (MAN)
    - Wide area network (WAN)
    - Wireless local area network (WLAN)
    - Personal area network (PAN)
    - Campus area network (CAN)
    - Storage area network (SAN)
    - Software-defined wide-area network (SDWAN)
    - Multiprotocol label switching (MPLS)
    - Multipoint generic routing encapsulation (mGRE)

        - Service-related entry point
            - Demarcation point
            - Smartjack

- Virtual network endpoints
    - vSwitch
    - Virtual network interface card (vNIC)
    - Network function virtualization (NFV)
    - Hypervisor
- Provider links
    - Satellite
    - Digital subscriber line (DSL)
    - Cable
    - Leased line
    - Metro-optical

What comes to mind when you think of a computer network? Is it the Internet? Is it email? Is it the wireless connection that lets you print to your printer from your laptop? Is it the smart thermostat and lights in your home? Whatever your current perception of a computer network, this chapter just might help you expand your thought process in this regard. Be aware that although you think of computer networks as interconnecting computers, today's computer networks interconnect a variety of devices in addition to just computers. Examples include game consoles, video-surveillance devices, IP-based telephones, tablets, and smartphones. Therefore, throughout this book, think of the term computer network as being synonymous with the more generic term network, because these terms are used interchangeably. The goal of this chapter is to acquaint you with the purpose of a network and help you categorize a given network based on criteria such as geography, topology, and the location of the network's resources.

Foundation Topics - Defining a Network

The movie *Field of Dreams* featured the statement "if you build it, it will come." This statement most certainly applies to the evolution of network-based services in modern-day networks. Computer networks are no longer relegated to allowing a group of computers to access a common set of files stored on a computer chosen as a *file server*. Instead, with the building of high-speed, highly redundant networks, network architects are seeing the wisdom of placing a variety of traffic types on a single network. Examples include voice and video, in addition to data. As you will learn in this chapter, the Internet of Things (IoT) means that just about everything wants to join your network, from the lights in your home to many of your household appliances. One could argue that a network is the sum of its parts. So, as you begin your study of networking, you should start to gain a basic understanding of fundamental networking components, including such entities as the client, server, hub, switch, and router, as well as the media used to interconnect these devices.

The Purpose of Networks

The basic purpose of a network is to make connections. These connections might be between a PC and printer or between a laptop and the Internet, as just a couple of examples. However, the true value of a network comes from the traffic flowing over those connections. Consider a sampling of applications that can travel over a network's connections:

- File sharing between two computers
- Video chatting between computers located in different parts of the world
- Surfing the Web (for example, to use social media sites, watch streaming video, listen to an Internet radio station, or do research for a school term paper)
- Instant messaging (IM) between computers with IM software installed
- Email
- Voice over IP (VoIP), to replace traditional telephony systems

A term given to a network transporting multiple types of traffic (for example, voice, video, and data) is a converged network. A converged network might offer significant cost savings to organizations that previously supported separate network infrastructures for voice, data, and video traffic. This convergence also potentially reduces staffing costs because only a single network needs to be supported, rather than separate networks for separate traffic types.

Network Types and Connections

As you might be sensing at this point, not all networks look the same. They vary in many ways. One criterion by which networks are classified is how geographically dispersed the network's components are. For example, a network might interconnect devices within an office, or a network might interconnect a database at a corporate headquarters location with a remote sales office on the opposite side of the globe. Based on the geographic dispersion of network components, you can classify networks into various categories, including the following:
<key_topic>

- Local area net work (LAN)
- Wide area network (WAN)
- Wireless local area network (WLAN)
- Storage area network (SAN)
- Campus area network (CAN) Metropolitan area network (MAN) Personal area network (PAN)

In addition to discussing these categories, this section also discusses SDWAN and mGRE technologies, which help create network *overlays*, which permit virtualization of network topologies. The "traditional" hardware and software of the network make up what is termed the *underlay*.

LAN A LAN interconnects network components within a local area (for example, within a building). Examples of common LAN technologies you are likely to meet include Ethernet (that is, IEEE 802.3) and wireless networks (that is, IEEE 802.11). Figure 2-1 illustrates an example of a LAN. Note IEEE stands for Institute of Electrical and Electronics Engineers, which is an internationally recognized standards body.

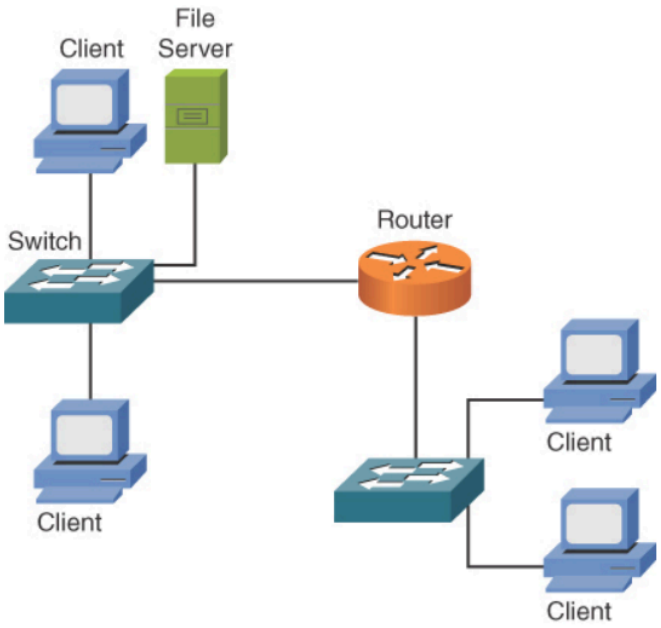


FIGURE 2-1 Sample LAN topology

WAN

A WAN interconnects network components that are geographically separated. For example, a corporate headquarters might have multiple WAN connections to remote office sites. Multiprotocol Label Switching (MPLS) and Asynchronous Transfer Mode (ATM) are examples of WAN technologies. Figure 2-2 depicts a simple WAN topology, which interconnects two geographically distinct locations.

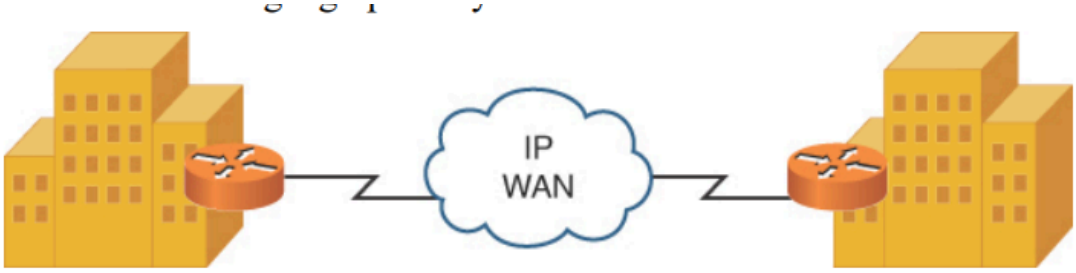


FIGURE 2-2 Sample WAN Topology

WLAN A local area network made up of wireless networking devices is a wireless local area network (WLAN). SAN You can construct a high-speed, highly reliable network for the express purpose of transmitting stored data. This network is called a storage area network (SAN). Other Categories of Networks Although LAN and WAN are the most common terms used to categorize computer networks based on geography, other categories include campus area network (CAN), metropolitan area network (MAN), and personal area network (PAN).

CAN The first time I discovered a CAN-type topology was at a major university. The university covered several square miles and had several dozen buildings. Many of these buildings were running individual LANs, and these building-centric LANs were interconnected. The interconnection of these LANs created another network type: a campus area network (CAN). Besides being common on university campuses, CANs are often used in industrial parks and business parks. MAN More widespread than a CAN and less widespread than a WAN, a metropolitan area network (MAN) interconnects locations scattered throughout a metropolitan area. Imagine, for example, that a business in Chicago has a location near O'Hare Airport, another location near the Navy Pier, and another location in the Willis Tower (previously known as the Sears Tower). If a service provider could interconnect those locations using a high-speed network, such as a 10Gbps (that is, 10 billion bits per second) network, the interconnection of those locations would form a MAN. One example of a MAN technology is Metro Ethernet, which features much higher speeds than the traditional WAN technologies that were used in the past to connect such locations.

PAN A personal area network (PAN) is a network whose scale is even smaller than a LAN. For example, a connection between a PC and a digital camera via a universal serial bus (USB) cable could be considered a PAN. Another example is a PC connected to an external hard drive via a USB 3.0 or Thunderbolt connection. A PAN, however, is not necessarily a wired connection. A Bluetooth connection between your cell phone and your car's audio system is considered a wireless PAN (WPAN). The main distinction of a PAN is that its range is typically limited to just a few meters. Software-Defined Wide Area Network (SD-WAN)

Almost every major networking vendor now offers a *software-defined wide area network (SD-WAN)* product. Cisco Systems, for example, currently has two of these solutions in its portfolio: the Cisco SD-WAN that Cisco acquired through the purchase of Viptela and the

SD-WAN that is part of Cisco's Meraki acquisition. An SD-WAN provides a simple policy and profile approach to managing the WAN. It also provides tools that enable new levels of visibility into and control over the use of the varied WAN circuits in the typical enterprise today.

Multiprotocol Label Switching (MPLS)

Multiprotocol label switching (MPLS) is growing popularity as a WAN technology used by service providers. This growth in popularity is due in part to MPLS's capability to support multiple protocols on the same network – for example, an MPLS network can accommodate users connecting via Frame Relay or ATM on the same MPLS backbone—and MPLS's capability to perform traffic engineering (which allows traffic to be dynamically routed within an MPLS cloud, based on current load conditions of specific links and availability of alternate paths). MPLS inserts a 32-bit header between Layer 2 and Layer 3 headers. Because this header is shimmed between the Layer 2 and Layer 3 headers, it is sometimes referred to as a shim header. Also, because the MPLS header resides between the Layer 2 and Layer 3 headers, MPLS is considered to be a Layer 2½ technology. The 32-bit header contains a 20-bit label that is used to make forwarding decisions within an MPLS cloud. The process of routing MPLS frames through an MPLS cloud is referred to as label switching. Figure 2-3 shows a sample MPLS network. Table 2-1 defines the various MPLS network elements shown in the figure.

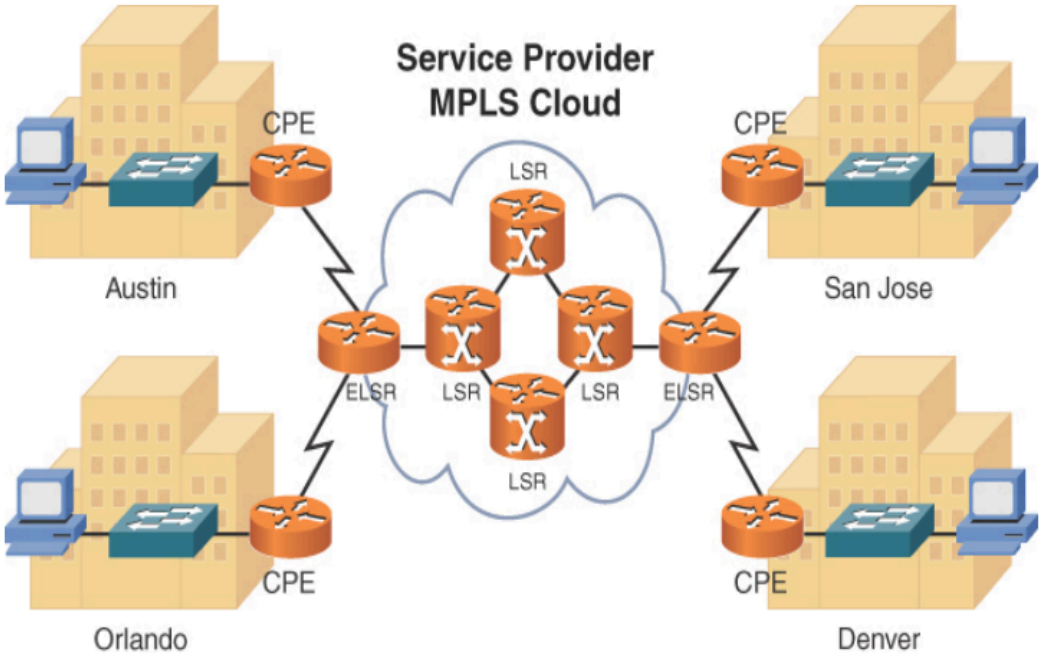


FIGURE 2-3 MPLS Sample Topology

Table 2-1 MPLS Network Elements

Element	Description
CPE	A customer premises equipment (CPE) device resides at a customer site. A router, for example, could be CPE that connects a customer with an MPLS service provider.
CE	A customer edge (CE) router is a customer router that provides the connectivity between the customer network and the service provider network. CE routers use static or dynamic routing protocols but do not run MPLS. The MPLS function occurs in the service provider network.

Element	Description
ELSR	An edge label switch router (ELSR) resides at the edge of an MPLS service provider's cloud and interconnects a service provider to one or more customers.
PE	A provider edge (PE) router is the MPLS service provider's router that connects to the customer router. PE is another name for ELSR.
LSR	A label switch router (LSR) resides as part of a service provider's MPLS cloud and makes frame-forwarding decisions based on labels applied to frames.
P	A provider (P) router is a service provider internal router that doesn't directly interface with the customer routers. A P router is internal to the service provider's network.

An MPLS frame does not maintain the same label throughout the MPLS cloud. Rather, an LSR receives a frame, examines the label on the frame, makes a forwarding decision based on the label, places a new label on the frame, and forwards the frame to the next LSR. This process of label switching is more efficient than routing based on Layer 3 IP addresses. The customer using a provider's network and the MPLS transport across that network is not normally aware of the details of the exact MPLS forwarding that is done by the service provider.

Multipoint Generic Routing Encapsulation (mGRE)

Normally, the flexible GRE encapsulation protocol is a point-to-point type of technology. Multipoint generic routing encapsulation (mGRE) variation of GRE is a multipoint technology. In fact, when you create a GRE tunnel, you specify source and destination interfaces for the tunnel. Technet24 |||||||||||||||||||||||||||||||||||||||||||||||| When you create an mGRE tunnel interface, all you specify is the source interface. You do not specify a destination because there are multiple, dynamic peers. mGRE is one of the key technologies in the Dynamic Multipoint VPN (DMVPN) solution. mGRE makes the tunnels that are formed dynamically between spokes and hubs (and between spokes and spokes). The mGRE tunnels are secured with Internet Protocol Security (IPsec) virtual private network (VPN) technology.

Networks Defined Based on Resource Location

Another way to categorize networks is based on where the network resources reside. For example, a *client/server* network is a collection of PCs all sharing files stored on a centralized server. However, if those PCs had their operating system (for example, Microsoft Windows 10 or macOS) configured for file sharing, they could share files from one another's hard drives. This is referred to as a *peer-to-peer* network because the peers (the PCs in this example) make resources available to other peers. The following sections describe client/server and peer-to-peer networks in more detail.

Client/Server Networks

Figure 2-4 illustrates an example of a *client/server* network, where a dedicated file server gives shared access to files, and a networked printer is available as a resource to the network's clients. Client/server networks are commonly used by businesses. Because resources are found on one or more servers, administration is simpler than administration of network resources on multiple peer devices.

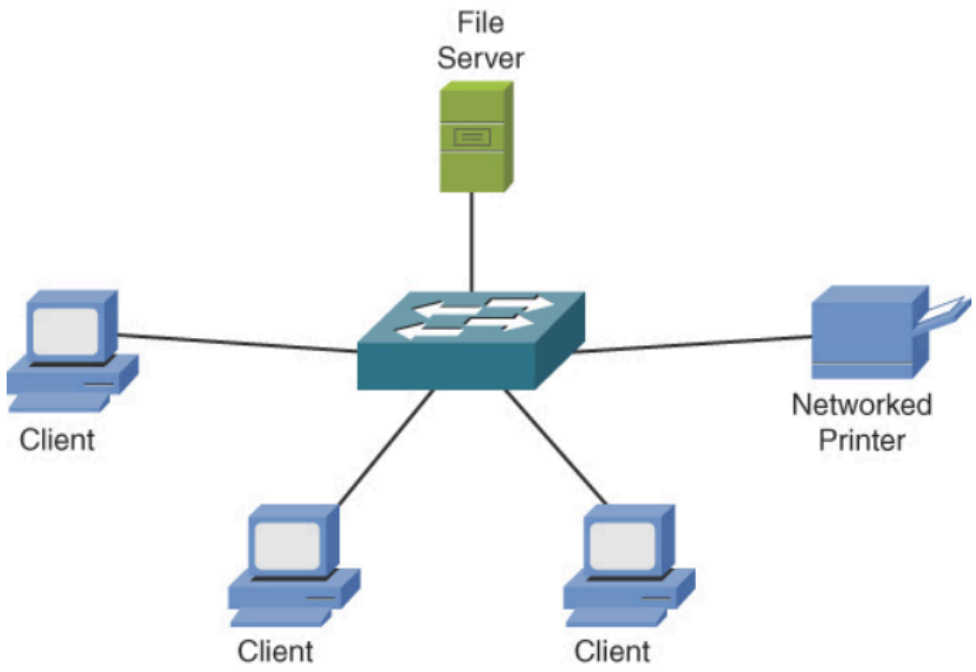


FIGURE 2-4 Client/Server Network Example

The performance of a client/server network can be better than that of a peerto-peer network because resources can be located on dedicated servers rather than on a PC running a variety of end-user applications. You can simplify backups because fewer locations must be backed up. However, client/server networks come with the extra expense of dedicated server resources. Table 2-2 lists the benefits and drawbacks of client/server networks.

Table 2-2: Characteristics, Benefits, and Drawbacks of Client/Server Networks

Networks

Characteristics	Benefits	Drawbacks
Client devices (for example, PCs) share a common set of resources (for example, file or print resources) located on one or more dedicated servers.	Client/server networks can easily scale, which might require the purchase of additional client licenses.	Because multiple clients might rely on a single server for their resources, the single server can become a single point of failure in the network.

Note

A server in a client/server network could be a computer running a network operating system (NOS) such as Linux Server or one of the Microsoft Windows Server operating systems. Alternatively, a server might be a host making its file system available to remote clients via the Network File Service (NFS) service, which was originally developed by Sun Microsystems.

Note

A variant of the traditional in the client/server network, where the server provides shared file access, is network-attached storage (NAS). A NAS device is a mass storage device that attaches directly to a network. Rather than running an advanced NOS, a NAS device usually makes files available to network clients via a service such as NFS.

Peer-to-Peer Networks

Peer-to-peer networks allow interconnected devices (for example, PCs) to share their resources with one another. Those resources could be, for example, files or printers. As an example of a peer-to-peer network, consider Figure 2-5, where each of the peers can share files on its own hard drive, and one of the peers has a directly attached printer that can be shared with the other peers in (on) the network.

with the other peers in the network.

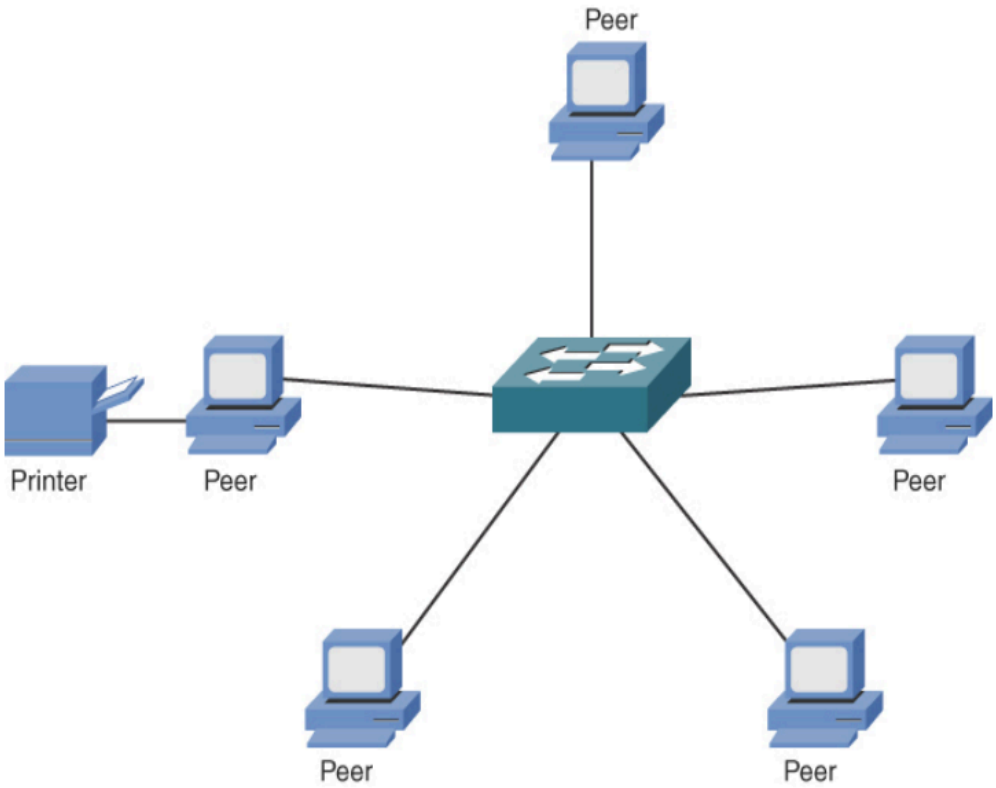


FIGURE 2-5 Peer-to-Peer Network Example

Peer-to-peer networks tend to be used in smaller businesses and in homes. The popularity of peer-to-peer networks is fueled in part by client operating systems that support file and print sharing. Scalability for peer-to-peer networks is a concern, however. Specifically, as the number of devices (that is peers, increases), the administration burden increases. For example, a network administrator might have to manage file permissions on multiple devices, as opposed to on a single server. Table 2-3 lists the characteristics, benefits, and drawbacks of peer-to-peer networks.

<key_topic>

Network

Characteristics	Benefits	Drawbacks
Client devices (for example, PCs) share their resources (for example, file and printer resources) with other client devices.	Peer-to-peer networks can be installed easily because resource sharing is made possible by the clients' operating systems, and knowledge of advanced networking operating systems is not required.	Scalability is limited because of the increased administration burden of managing multiple clients.
Resource sharing is made available through the clients' operating systems.	Peer-to-peer networks usually cost less than client/server networks because there is no requirement for dedicated server resources or advanced NOS software.	Performance might not be as strong as in a client/server network because the devices providing network resources might be performing other tasks not related to resource sharing (for example, word processing).

Note

Some networks have characteristics of both peer-to-peer and client/server networks. For example, all PCs in a company might point to a centralized server for accessing a shared database in a client/server topology. However, these PCs might simultaneously share files and printers with one another in a peer-to-peer topology. Such a network, which has a mixture of client/server and peer-to-peer characteristics, is called a *hybrid* network.

Networks Defined by Topology

In addition to classifying networks based on the geographic placement of their components, another approach to classifying a network is to use the network's topology. Looks can be deceiving, however. You need to be able to distinguish between a physical topology and a logical topology.

Physical Versus Logical Topology

Even if a network appears to be a star topology (that is, where the network components all connect to a centralized device, such as a switch), the traffic might be flowing in a circular pattern through all the network components attached to the centralized device. The actual traffic flow determines the *logical topology*, whereas the way components are physically interconnected determines the *physical topology*. For example, consider Figure 2-6, which shows a collection of computers connected to a Token Ring media access unit (MAU). From a quick inspection of Figure 2-6, you can conclude that the devices are physically connected in a star topology, where the connected devices radiate out from a centralized aggregation point (the MAU in this example).

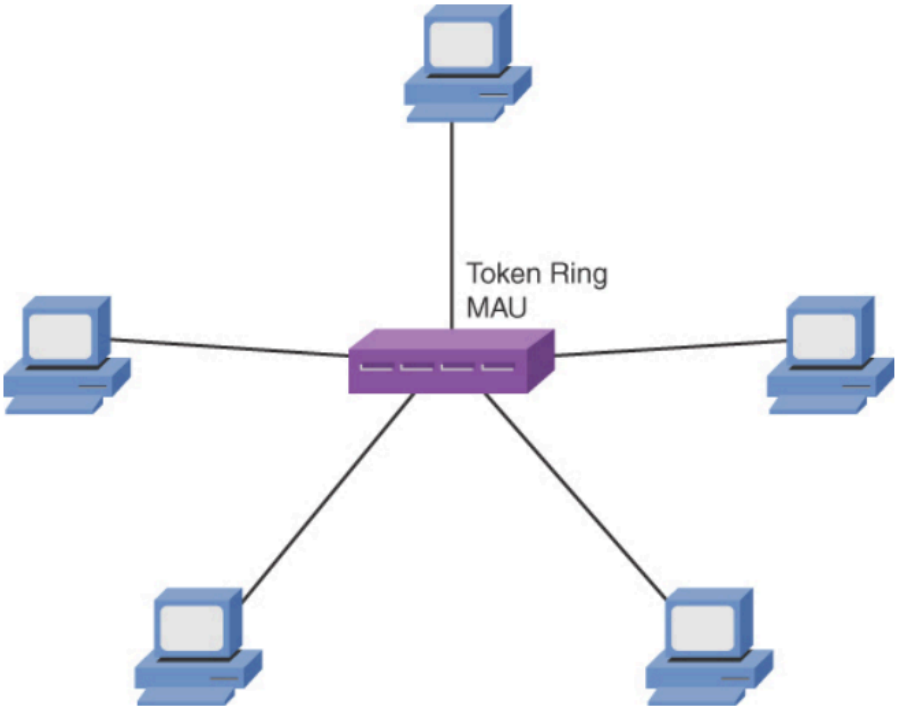


FIGURE 2-6 Physical Star Topology

Now contrast the physical topology in Figure 2-6 with the logical topology illustrated in Figure 2-7. Although you can see the computers physically connect to a centralized MAU, when you examine the flow of traffic through (or in this case, around) the network, you see the traffic flow actually loops around and around the network. The traffic flow dictates how to classify a network's logical topology. In this instance, the logical topology is a *ring topology* because the traffic circulates around the network as if circulating around a ring.

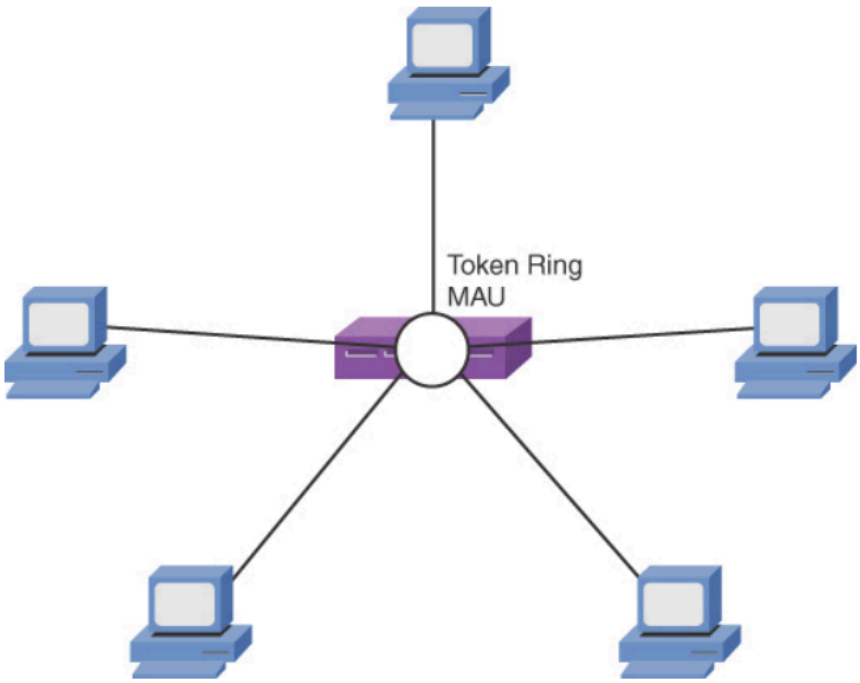


FIGURE 2-7 Logical Ring Topology

Note

Before you run out and try to purchase a Token Ring for your network for your LAN, keep in mind you'll only see this networking technology in museums now!

Bus Topology

A bus topology, as depicted in Figure 2-8, typically has a cable running through the area that requires connectivity, and devices that need to connect to the network tap into this cable. Early Ethernet networks relied on bus topologies. A network tap might be in the form of a T connector (used in older 10BASE2 networks) or a vampire tap (used in older 10BASE5 networks).

Figure 2-9 shows an example of a T connector.

Figure 2-9 shows an example of a T connector.

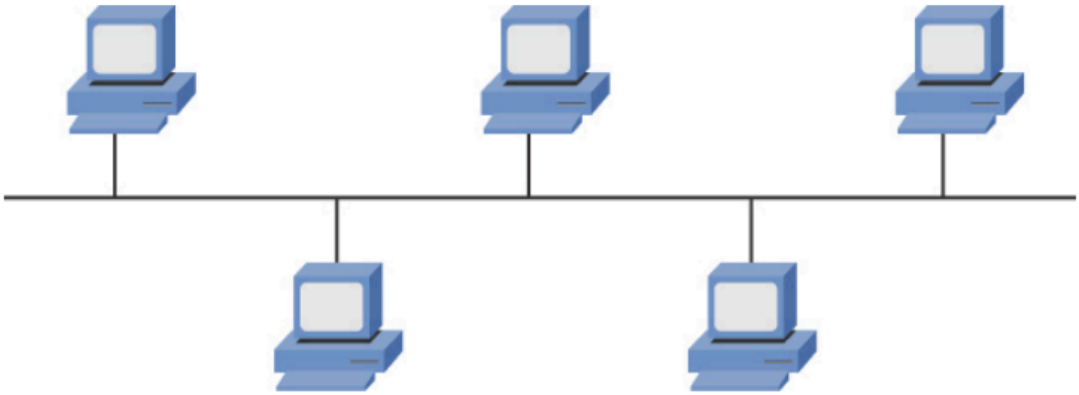


FIGURE 2-8 Bus Topology

FIGURE 2-9 T Connector

A bus and all devices connected to that bus make up a *network segment.* A single network segment is a single collision domain, which means that all devices connected to the bus might try to gain access to the bus at the same time, resulting in an error condition known as a

collision. Table 2-4 shows some of the characteristics, benefits, and drawbacks of a bus topology.

Table 2-4 Characteristics, Benefits, and Drawbacks of a Bus Topology

Characteristics	Benefits	Drawbacks
One cable is used per network segment.	Less cable is needed to install a bus topology than is required with other topologies.	Because a single cable is used per network segment, the cable is potentially a single point of failure.
To support appropriate electrical characteristics of the cable, the cable requires a terminator (of a specific resistance) at each end of the cable.	Depending on the media used by the bus, a bus topology can be less expensive than other topologies.	Troubleshooting a bus topology can be difficult because problem isolation might require inspection of multiple network taps to make sure they either have a device connected or are properly terminated.
Bus topologies were popular in early Ethernet networks.	Installation of a network based on a bus topology is easier than with some other topologies, which might require extra wiring to be installed.	Adding devices to a bus might cause an outage for other users on the bus.

Characteristics	Benefits	Drawbacks
Network components tap directly into the cable via a connector such as a T connector or a vampire tap.	—	An error condition existing on one device on the bus can affect performance of other devices on the bus.
—	—	A bus topology does not scale well because all devices share the bandwidth available on the bus. Also, if two devices on the bus simultaneously request access to the bus, an error condition results.

Ring Topology

Figure 2-10 provides an example of a *ring topology*, where traffic flows in a circular fashion around a closed loop (that is, a ring). Typically, a ring topology sends data, in a single direction, to each connected device in turn, until the intended destination receives the data. Token Ring networks relied on a ring topology.

Token Ring was not the only popular ring-based topology popular in networks in the 1990s. Fiber Distributed Data Interface (FDDI) was another variant of a ring-based topology. Most FDDI networks (which, as the name suggests, have fiber optics as the media) used just not one ring but two. These two rings sent data in opposite directions, resulting in *counter-rotating* rings. One benefit of counter-rotating rings was that if a fiber broke, the stations on each side of the break could interconnect their two rings to create a single ring capable of reaching all stations on the ring. Because a ring topology allows devices to take turns transmitting on the ring, contention for media access was not a problem, as it was for a bus topology. If a network had a single ring, however, the ring was potentially a single point of failure. If the ring was broken at any point, data stopped flowing. Table 2-5 lists some of the primary characteristics, benefits, and drawbacks of a ring topology.

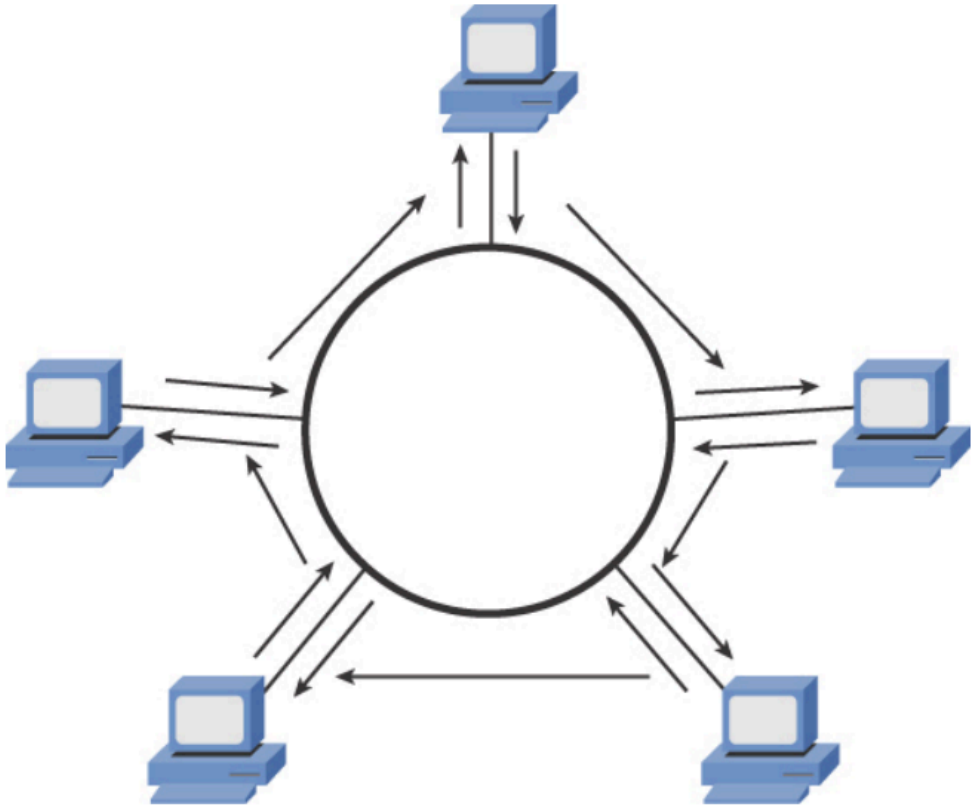


FIGURE 2-10 Ring Topology

<key_topic>

Table 2-5 Characteristics, Benefits, and Drawbacks of a Ring Topology

Characteristics	Benefits	Drawbacks

Characteristics	Benefits	Drawbacks
Devices are interconnected by connecting to a single ring or, in some	A dual-ring topology adds a layer of fault tolerance. Therefore, if a cable break occurs, connectivity to all devices can be restored.	A break in a ring when a single ring topology is used results in a network outage for all devices connected to the ring.

Each device on a ring includes both a receiver (for the incoming cable) and a transmitter (for the outgoing cable).	Troubleshooting is simplified in the event of a cable break because each device on a ring contains a repeater. When the repeater on the far side of a cable break does not receive any data within a certain amount of time, it reports an error condition, typically in the form of an indicator light on a network interface card (NIC).	Rings have scalability limitations. Specifically, a ring has a maximum length and a maximum number of attached stations. Once either of these limits is exceeded, a single ring might need to be divided into two interconnected rings. A network maintenance window might need to be scheduled to perform this ring division.

Characteristics	Benefits	Drawbacks
Each device on the ring repeats the signal it receives.	—	Because a ring must be a complete loop, the amount of cable required for a ring is usually higher than the amount of cable required for a bus topology serving the same number of devices.

Star Topology

Figure 2-11 shows a sample *star topology* with a hub at the center of the topology and a collection of clients at the center of the hub. Notice that a star topology has a central point from which all attached devices radiate. In LANs in the early 1990s, that centralized device was typically a hub. Modern networks, however, usually have a switch located at the center of the star.

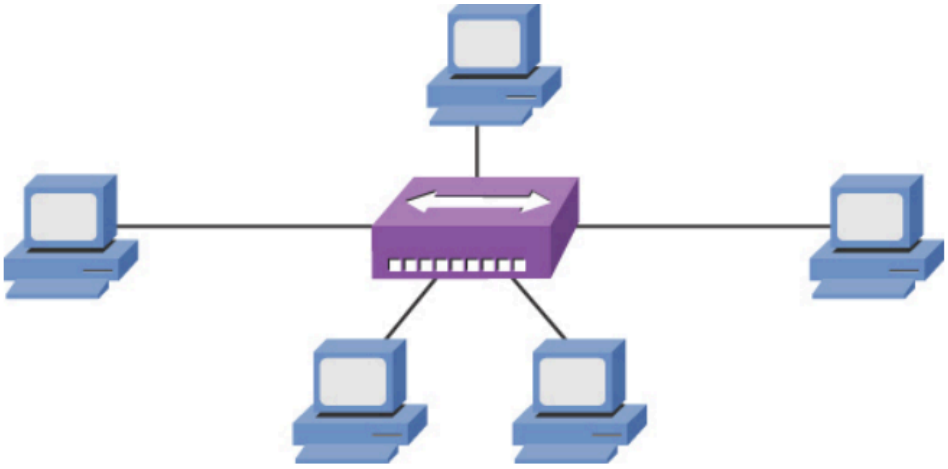


FIGURE 2-11 Star Topology

The star topology is the most popular physical LAN topology in use today, with an Ethernet switch at the center of the of the star and unshielded twisted-pair (UTP) cable used to connect from the switch ports to the clients. Table 2-6 identifies some of the primary characteristics, benefits and drawbacks of a star topology.

Table 2-6 Characteristics, Benefits, and Drawbacks of a Star Topology

Characteristics	Benefits	Drawbacks
Devices have independent connections to a central device (for example, a hub or a switch).	A cable break impacts only the device connected via the broken cable and not the entire topology.	More cable is required for a star topology than for bus or ring topologies because each device requires its own cable to connect back to the central device.
Star topologies are commonly used with Ethernet technologies (described in Chapter 3).	Troubleshooting is relatively simple because a central device in the star topology acts as the aggregation point for all the connected devices.	Installation can take longer for a star topology than for a bus or ring topology because more cable runs must be installed.

Hub-and-Spoke Topology

When interconnecting multiple sites (for example, multiple corporate locations) via WAN links, a *hub-and-spoke topology* may be used, with a WAN link from each remote site (that is, *spoke site*) to the main site (that is, the *hub site*). This approach, as an example of what is shown in Figure 2-12, is similar to the star topology used in LANs.

With WAN links, a service provider is paid a recurring fee for each link. Therefore, a hub-and-spoke topology helps minimize WAN expenses by not |||||||||||||||||||||||||||||||||||||||||||| directly connecting any two spoke locations. If two spoke locations need to communicate with each other, their communication is sent via the hub location. Table 2-7 describes the characteristics, benefits, and drawbacks of a hub-and-spoke WAN topology.

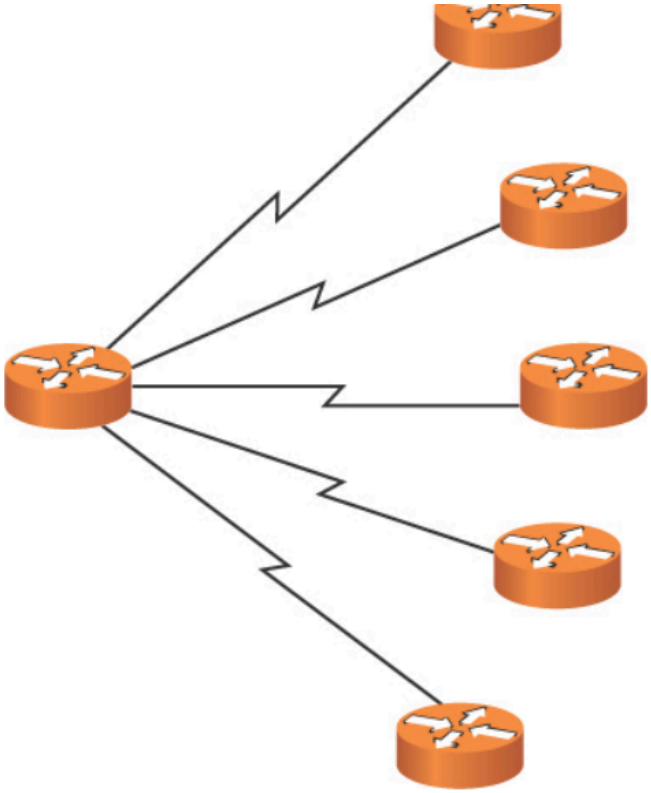


FIGURE 2-12 Hub-and-Spoke Topology

<key_topic>

Table 2-7 Characteristics, Benefits, and Drawbacks of a Hub-and-Spoke WAN Topology

Characteristics	Benefits	Drawbacks
Each remote site (that is, a spoke) connects to a main site (that is, the hub) via a WAN link.	Costs are reduced (as compared to with a full-mesh or partial-mesh topology) because a minimal number of links is used.	Suboptimal routes must be used between remote sites because all intersite communication must travel via the main site.
Communication between two remote sites travels through the hub site.	Adding one or more additional sites is easy (compared to in a full-mesh or partial-mesh topology) because only one link needs to be added per site.	Because all remote sites converge on the main site, this hub site is potentially a single point of failure.
—	—	Because each remote site is reachable by only a single WAN link, the hub-and-spoke topology lacks redundancy.

Full-Mesh Topology

Whereas a hub-and-spoke topology lacks redundancy and suffers from suboptimal routes, a *full-mesh topology*, as shown in Figure 2-13, directly connects each site to each other site.

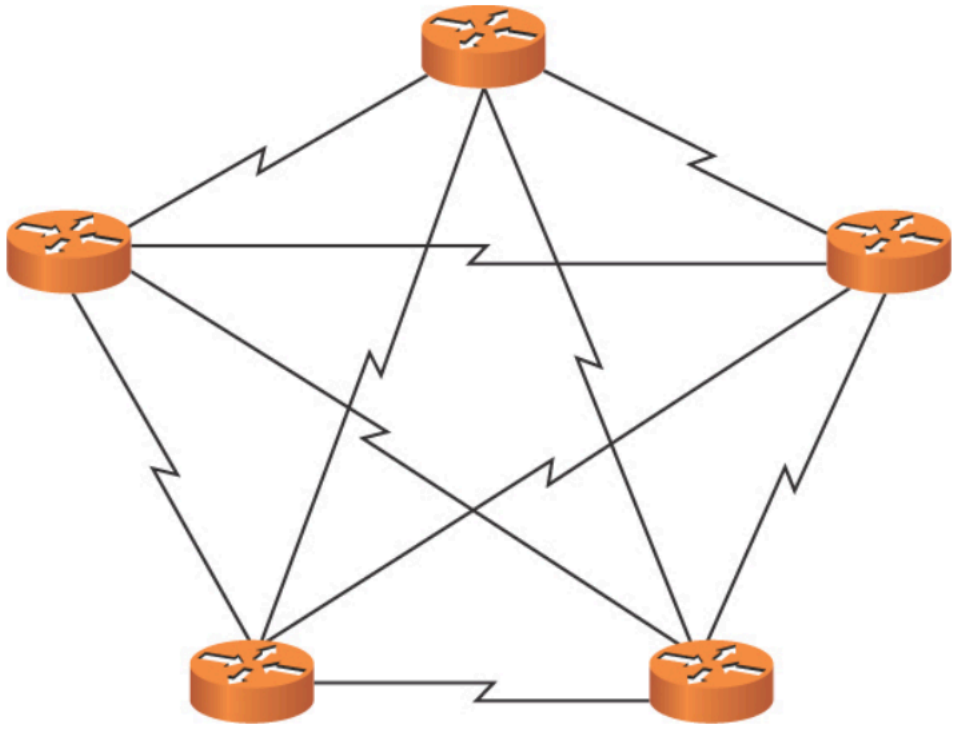


FIGURE 2-13 Full-Mesh Topology

Because each site connects directly to every other site, an optimal path can be selected, as opposed to relaying traffic via another site. Also, a full-mesh topology is highly fault tolerant. By inspecting Figure 2-13, you can see that multiple links in the topology could be lost, and every site might still be able to connect to every other site. Table 2-8 summarizes the characteristics, benefits, and drawbacks of a full-mesh topology.

Table 2-8 Characteristics, Benefits, and Drawbacks of a Full-Mesh WAN Topology

Characteristics	Benefits	Drawbacks
Every site has a direct WAN connection to every other site.	An optimal route exists between any two sites.	A full-mesh network can be difficult and expensive to scale because the addition of one new site requires a new WAN link between the new site and every other existing site.

		site.
The number of required WAN connections can be calculated with the formula $w = n \times (n - 1) / 2$, where w = the number of WAN links and n = the number of sites. For example, a network with 10 sites would require 45 WAN connections to form a fully meshed network: $45 = 10 \times (10 - 1) / 2$.	A full-mesh network is fault tolerant because one or more links can be lost, and reachability between all sites might still be maintained.	—
—	Troubleshooting a full-mesh network is relatively easy because each link is independent of the other links.	—

Partial-Mesh Topology

A partial-mesh WAN topology, as depicted in Figure 2-14, is a hybrid of the previously described hub-and-spoke topology and full-mesh topology. Specifically, a *partial-mesh topology* can be designed to offer an optimal |||||||||||||||||||||||||||||||||||||||||||| route between selected sites while avoiding the expense of interconnecting every site to every other site.

When designing a partial-mesh topology, a network designer must consider network traffic patterns and strategically add links interconnecting sites that have higher volumes of traffic between themselves. Table 2-9 highlights the characteristics, benefits, and drawbacks of a partial-mesh topology.

characteristics, benefits, and drawbacks of a partial-mesh topology.

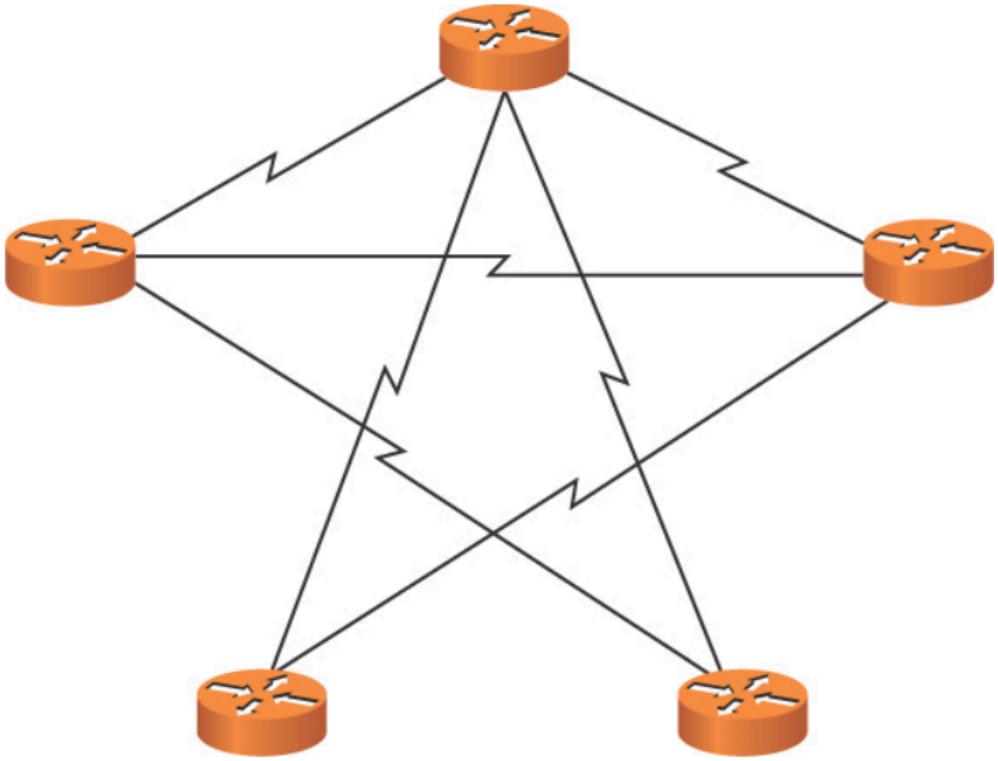


FIGURE 2-14 Partial-Mesh Topology

<key_topic>

Table 2-9 Characteristics, Benefits, and Drawbacks of a Partial-Mesh Topology

Characteristics	Benefits	Drawbacks
Selected sites (that is, sites with frequent intersite communication) are interconnected via direct links, whereas sites that have less-frequent communication can communicate via another site.	A partial-mesh topology provides optimal routes between selected sites with higher intersite traffic volumes while avoiding the expense of interconnecting every site to every other site.	A partial-mesh topology is less fault tolerant than a full-mesh topology.
A partial-mesh topology uses fewer links than a full-mesh topology and more links than a hub-and-spoke topology for interconnecting the same number of sites.	A partial-mesh topology is more redundant than a hub-and-spoke topology.	A partial-mesh topology is more expensive than a hub-and-spoke topology.

Note

There are plenty of network topologies today that use a variety of different design approaches. This often results in several approaches being used in one solution. We often term this a *hybrid* approach.

Service-Related Entry Points

An important aspect of the network topology is the *service-related entry point.* Two possibilities of which you should be aware for the Network+ exam are the demarcation point and the smartjack.

A *demarcation point* (also known as a *demarc* or a *demarc extension*) is the point in a telephone network where the maintenance responsibility passes from a telephone company to the subscriber (unless the subscriber has ||||||||||||||||||||||||||||||||||||||||| purchased inside wiring maintenance). This demarc is typically located in a box mounted to the outside of a customer's building (for example, a residential home). This box is called a *network interface device (NID)*.

A *smartjack* is a type of networking interface that adds circuitry. This circuitry adds such features as converting between framing formats on a digital circuit (for example, a T1 circuit), supporting remote diagnostics, and regenerating a digital signal.

Virtual Network Concepts

A major data center paradigm shift is underway. This shift is away from a company having its own data center with its raised flooring and large air conditioning system) containing multiple physical servers, each of which offer a specific service (for example, email, DNS services, or Microsoft Active Directory).

Virtual Servers

The computing power available in a single high-end server is often sufficient to handle the tasks of multiple independent servers. With the advent of virtualization, multiple servers (which might be running different operating systems) can run in virtual server instances on one physical device. For example, a single high-end server might be running an instance of a Microsoft Windows Server providing Microsoft Active Directory (AD) services to an enterprise, while simultaneously running an instance of a Linux server acting as a corporate web server, and at the same time acting as a Oracle Solaris UNIX server providing corporate DNS services. Figure 2-15 illustrates the concept of a virtual server. Although the virtual server in the figure uses a single NIC to connect to an Ethernet switch, many virtual server platforms support multiple NICs. Having multiple NICs offers increased throughput and load balancing.

<key_topic>

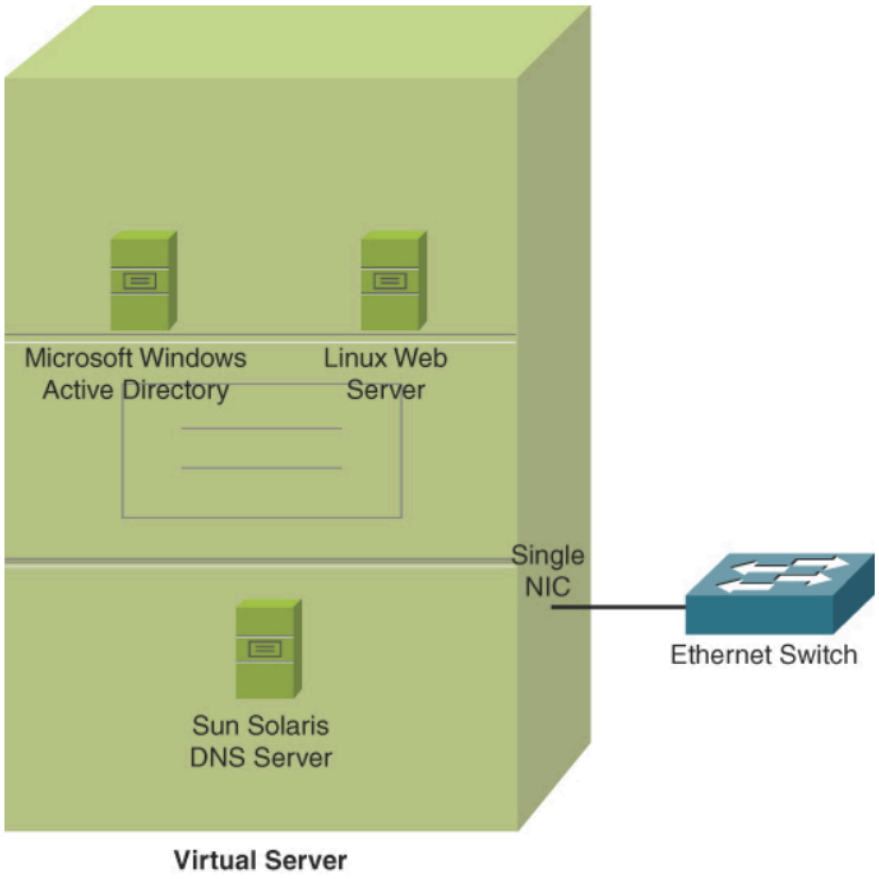


FIGURE 2-15 Virtual Server

Note

Although the example in Figure 2-15 uses a Linux-based web server, web servers can run on a variety of operating system platforms. For example, Microsoft Windows servers support a web server application called *Internet Information Services (IIS)*, which was previously known as Internet Information Server.

Virtualization is possible with servers thanks to specialized software called a hypervisor. The *hypervisor* takes physical hardware and abstracts it for the virtual server. The extent of virtualization is amazing, and even the NIC of each virtual server can be represented virtually (*virtual network interface card[vNIC]*). Figure 2-16 shows some of the configuration options for a vNIC in a virtualized environment. Notice that technologies such as VLANs and QoS (in this case, bandwidth management) are still possible in the virtualized world.

possible in the virtualized world.

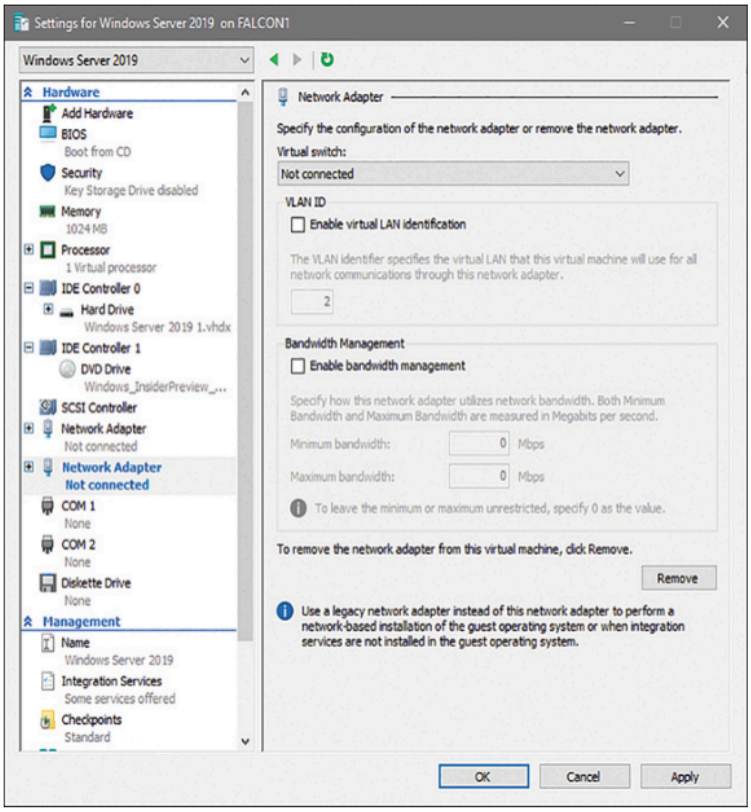


FIGURE 2-16 Configuring a vNIC

The networks and systems supporting virtual servers also commonly have network-attached storage (NAS), where disk storage is delivered as a service over the network. A technology for network storage is IP-based Small Computer System Interface (iSCSI). With iSCSI, a client using the storage is referred to as an *initiator*, SCSI). With iSCSI, a client using the storage is referred to as an initiator, and the system providing the iSCSI storage is called the iSCSI *target* The networks supporting iSCSI are often configured to support larger-than-normal frame sizes, referred to as *jumbo frames*. Fibre Channel is another technology that can deliver storage services over a network. Thanks to high-speed Ethernet options today, you can even configure Fibre Channel over Ethernet (FCoE) to run a unified network for your SAN and non-storage-data traffic.

Note

A less commonly encountered communication technology, InfiniBand (IB), permits high-speed, low-latency communications between supercomputers.

Virtual Routers and Firewalls

Most of the vendors that create physical routers and firewalls also include an offering that includes virtualized routers and firewalls. The benefit of using a virtualized firewall or router is

that the same features of routing and security can be made in a virtual environment as in the physical environment. As part of interfacing with virtual networks, virtual network adapters can be used. For connectivity between the virtual world and the physical one, physical interfaces can be used to connect to the logical virtual interfaces. Virtualization is pervasive in networking today. In fact, virtualization is now used to implement all network functionality. There is actually a term for this approach: *network function virtualization (NFV)*. NFV can include security, storage, compute, and monitoring services.

Virtual Services (vSwitches)

One potential trade-off you make with the previously described virtual server scenario is that all servers belong to the same IP subnet, which could have Quality of Service (QoS) and security implications. If these server instances ran on separate physical devices, they could be attached to different ports on an Ethernet switch. These switch ports could belong to different VLANs, which could place each server in a different broadcast domain. Fortunately, some virtual servers allow you to still have Layer 2 control (for example, VLAN separation and filtering). This Layer 2 control is made possible by the virtual server not only virtualizing instances of servers but also virtualizing a Layer 2 switch. Figure 2-17 depicts a virtual switch (*vSwitch*). Notice that servers reside logically on separate VLANs, and frames from those servers are appropriately tagged when traveling over a trunk to the attached Ethernet switch. Figure 2-18 shows just how easy it is to configure a virtual switch in a network today. This is an example of the Hyper-V management software made available by Microsoft.

Virtual Desktops

<key_topic>

Another emerging virtualization technology is virtual desktops. Today's users are more mobile than ever before, and they need access to information traditionally stored on their office computers' hard drives from a variety of other locations. For example, a user might be at an airport using their smartphone and need access to a document they created on their office computer. With virtual desktops, a user's data is stored in a data center rather than on an office computer's hard drive. By providing authentication credentials, the user can establish a secure connection between the centralized repository of user data and their device, as shown in Figure 2-19, thus allowing the user to remotely access the desired document.

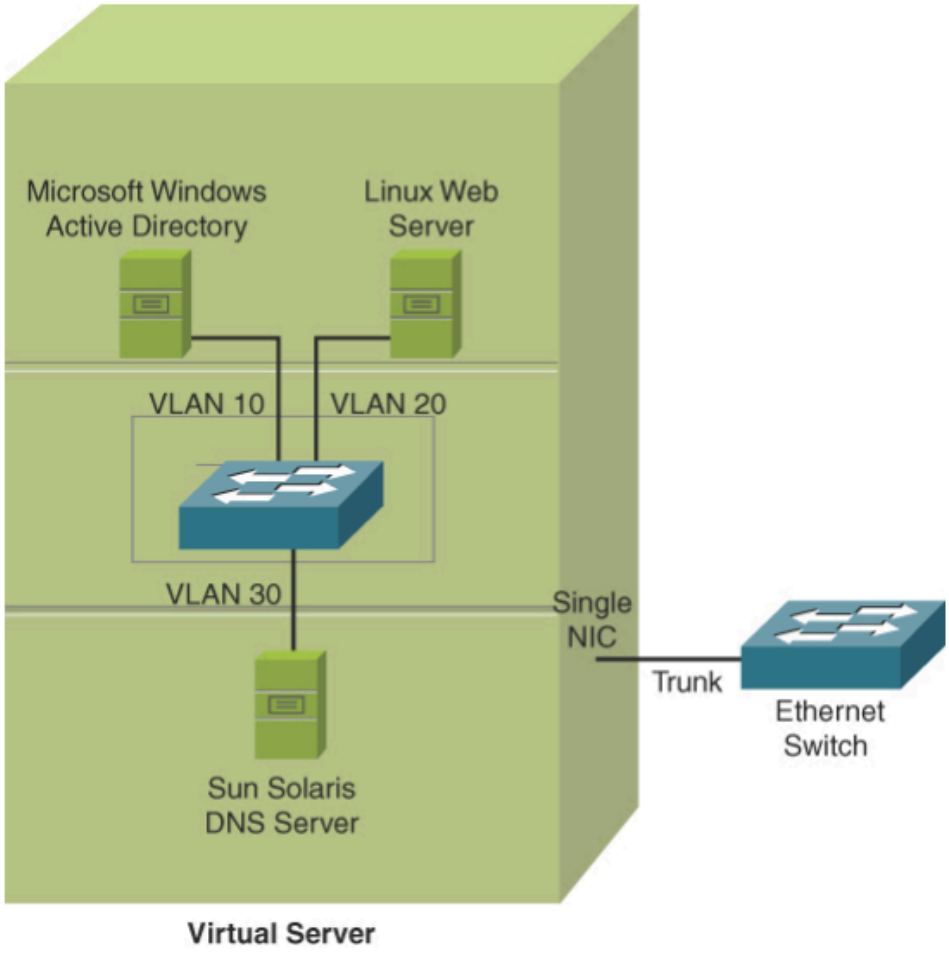


FIGURE 2-17. Virtual Server with a vSwitch

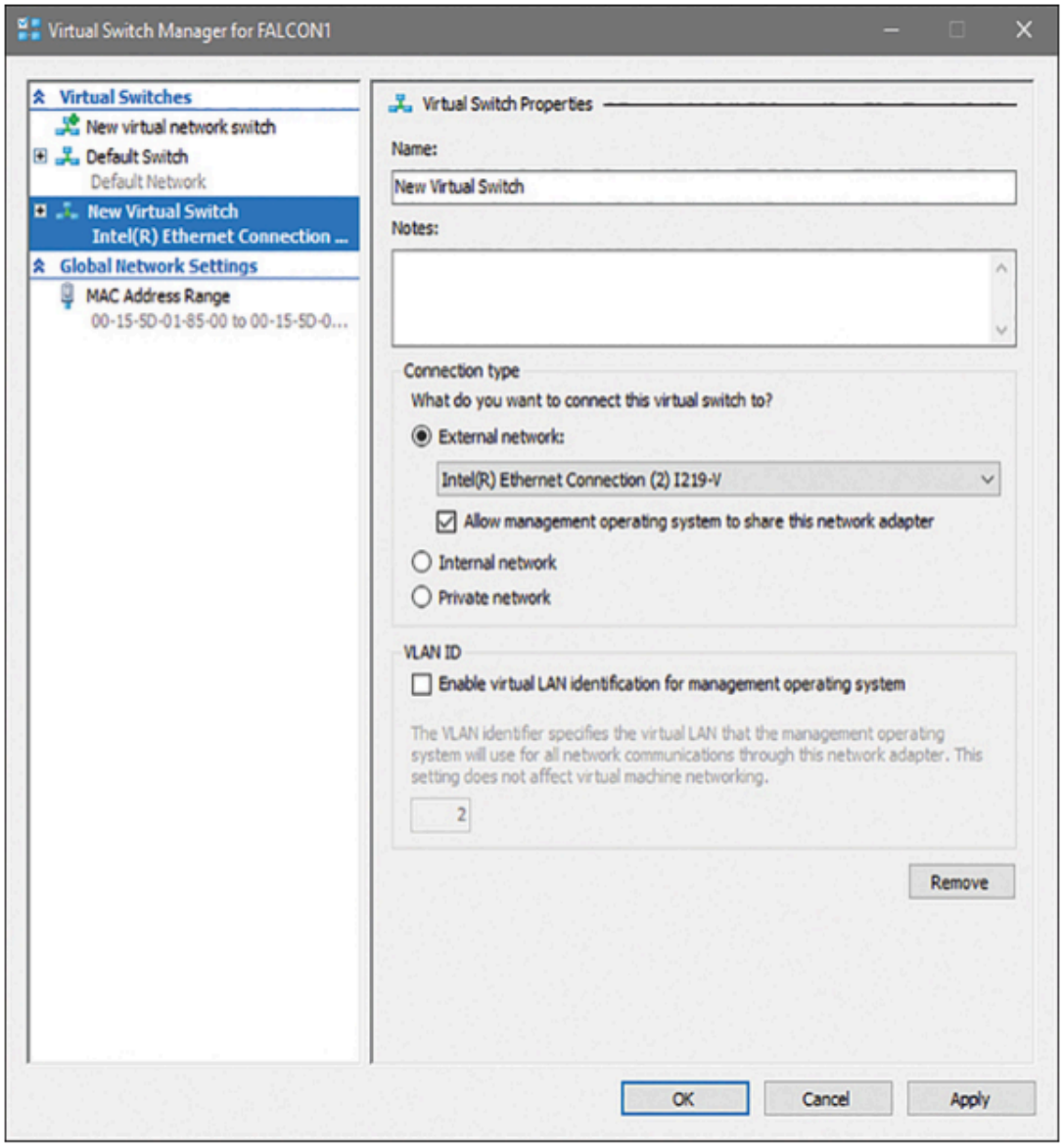


FIGURE 2-18 Configuring a Virtual Switch

<key_topic>

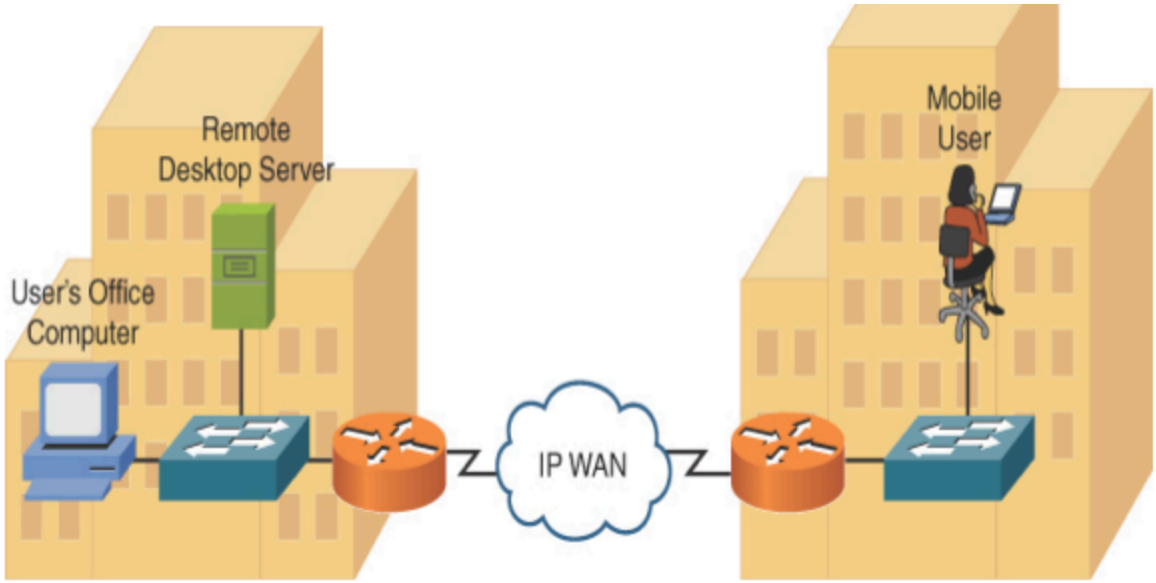


FIGURE 2-19 Virtual Desktop Topology

Other Virtualization Solutions

Although the previously discussed technologies – that is, virtual servers, virtual switches, and virtual desktops – are described as residing at a corporate location (that is, *on-site*), some service providers offer *off-site* options. Specifically, if a service provider's customer did not want to house and maintain its own data center, these virtualization technologies could be located at a service provider's data center, and the customer could be billed based on usage patterns. Such a service provider offering is called *network as a service (NaaS)*, implying that network features can be accessed by a service provider, just as a telephony service provider offers access to the public switched telephony network (PSTN), and an ISP offers access to the Public internet.

Provider Links

One type of network categorization is based on the provider link. There are many options here, including satellite, cable, DSL, and SONET.

Satellite

Many rural locations lack the option of connecting to an IP WAN or to the Internet via physical media (for example, a DSL modem or a broadband cable modem connection). For such locations, a *satellite* WAN connection, as illustrated in Figure 2-20, might be an option.

<key_topic>

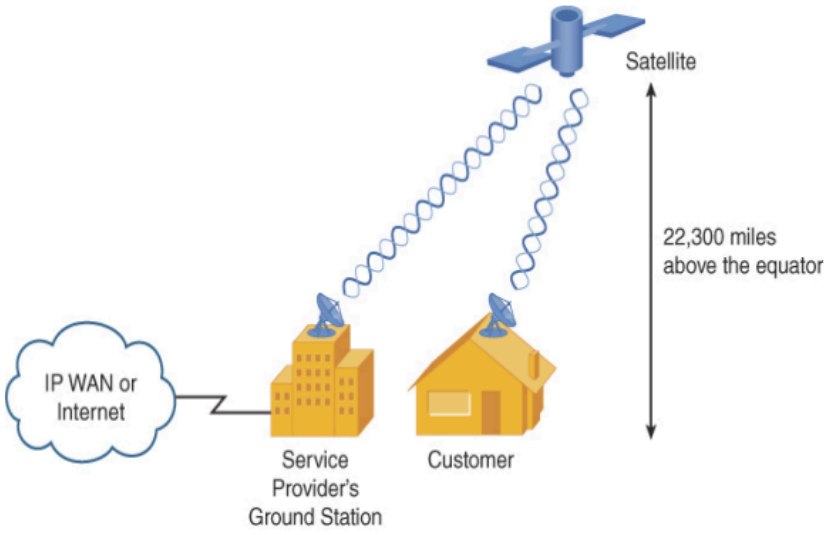


FIGURE 2-20 Satellite WAN Sample Topology

Most satellites used for WAN connectivity are in orbit above the earth's equator, about 22,300 miles high. Therefore, a customer in North America, for example, with a clear view of the southern sky would be able to install a satellite dish and establish a line-of-sight communication path with the orbiting satellite. The satellite would then relay transmissions back and forth between the customer's site and the service provider's ground station. The ground station could then provide connectivity, via physical media, to an IP WAN or to the Internet.

Two significant design considerations need to be taken into account:

- **Delay:** Radio waves travel at the speed of light, which is 186,000 miles per second, or 3x10^8 meters per second. This speed is specifically the speed of light (and radio waves) in a vacuum; however, for the purposes of this discussion, assume that these known values, even though, technically, the speed of light (and radio waves) is a bit slower when traveling through air than when traveling through a vacuum. Although these are fast speeds, consider the distance between a customer and the satellite. If a customer were located 2000 miles north of the equator, the approximate distance between the customer site and the satellite could be calculated using the Pythagorean theorem: d 2 = 20002 + 22,3002 .Solving the equation for d, which is the distance between the customer and the satellite, yields a result of approximately 22,390 miles. A transmission from a customer to a destination on the Internet (or an IP WAN) would have to travel from the customer to the satellite, from the satellite to the ground station, and then out to the Internet (or IP WAN). The propagation delay alone introduced by bouncing a signal off the satellite is approximately 241 ms—that is, (22,390 × 2) / 186,000 = .241 seconds = 241 ms). In addition, there are other delay components, such as processing delay (by the satellite and other networking devices), making the one-way delay greater than one-fourth of a second and, therefore, the round-trip delay greater

than one-half of a second. Such delays are not conducive to latency-sensitive applications such as voice over IP (VoIP).

- **Sensitivity to weather conditions:** Because communication between a customer's satellite dish and an orbiting satellite must travel through the Earth's atmosphere, weather conditions can impede communications. For example, if a thunderstorm is near the customer location, that customer might temporarily lose communication with their satellite.

Based on these design considerations, even though satellite WAN offers tremendous flexibility in terms of geographical location, more terrestrial-based solutions are preferred.

Digital Subscriber Line

A provider link that used to be commonplace in many residential and small business locations is digital subscriber line (DSL). DSL installations are becoming more and more rare, but they do still exist. For example, I still rely on one that acts as a backup WAN circuit in my small office/home office (SOHO) environment.

DSL is a group of technologies that provide high-speed data transmission over existing telephone wiring. DSL has several variants that differ in data rates and distance limitations:

<key_topic>

- **Asymmetric DSL (ADSL):** ADSL is a popular Internet-access solution for residential locations. Figure 2-21 shows a sample ADSL topology. Note that ADSL allows an existing telephone line to share the same line used for data for simultaneous voice and data. Also notice in Figure 2-21 that the maximum distance from a DSL modem to a DSL access multiplexer (DSLAM) is 18,000 feet. This limitation stems from a procedure telephone companies have used for decades to change the impedance of telephone lines. Here is a brief history: If wires in a telephone cable run side-by-side for several thousand feet, capacitance builds up in the line (which can cause echo). To counteract this capacitance, after 18,000 feet of cable, telephone companies insert a *load coil*, which adds inductance to the line. Electrically speaking, inductance is the opposite of capacitance. So, by adding a load coil, much of the built-up capacitance in a telephone cable is reduced. However, ADSL signals cannot cross a load coil, so there is a 18,000-foot distance limitation for ADSL.

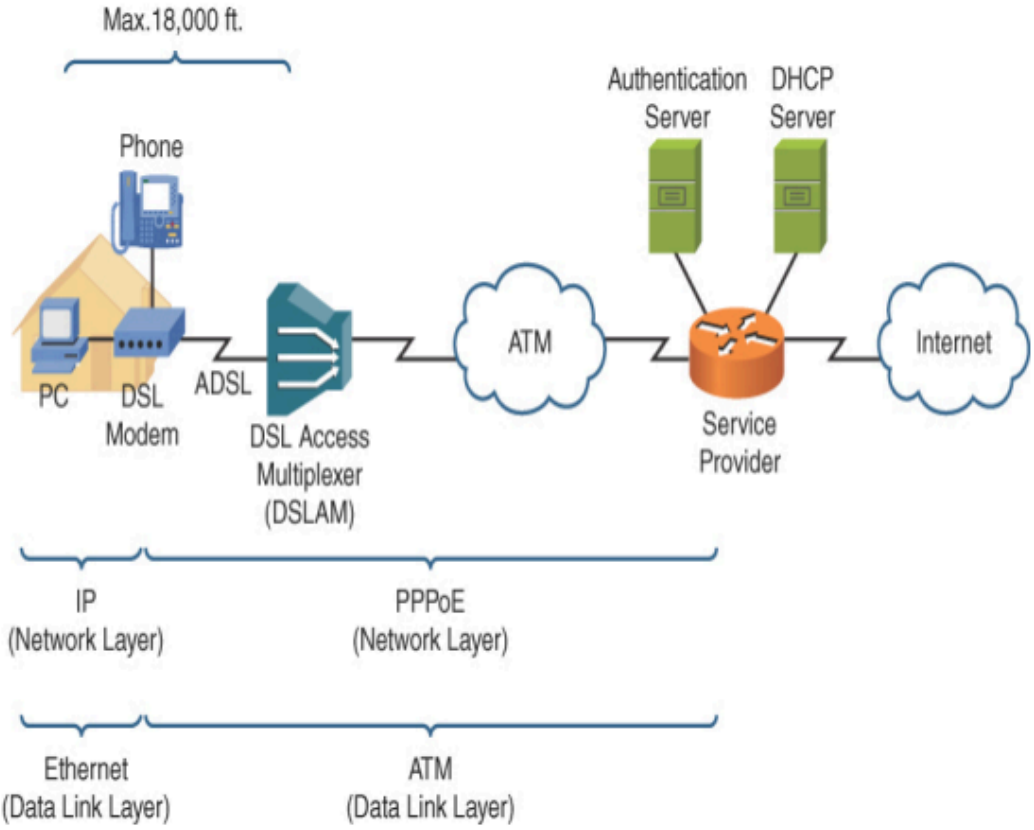


FIGURE 2-21 ADSL Sample Topology

Figure 2-21 also shows how a telephone line leaving a residence terminates on a DSLAM. A DSLAM acts as an aggregation point for multiple connections, and it connects via an ATM network back to a service provider's router. The service provider authenticates user credentials, given via Point-to-Point Protocol over Ethernet (PPPoE), using an authentication server. Also, the service provider has a DHCP server to hand out IP address information to end-user devices (for example, a PC or a wireless router connected to a DSL modem). The term *asymmetric* in asymmetric DSL indicates that the upstream and downstream speeds can be different. Typically, downstream speeds are greater than upstream speeds in an ADSL connection. The theoretical maximum downstream speed for an ADSL connection is 8Mbps, and the maximum upstream speed is 1.544Mbps (the speed |||||||||||||||||||||||||||||||||||||||||||| of a T1 circuit).

- **Symmetric DSL (SDSL):** Whereas ADSL has asymmetric (unequal) upstream and downstream speeds, by definition, SDSL has symmetric (equal) upstream and downstream speeds. Another distinction between ADSL and SDSL is that SDSL does not allow simultaneous voice and data on the same phone line. Therefore, SDSL is less popular in residential installations because an additional phone line is required for data. Although service providers vary, a typical maximum upstream/downstream rate is

1.168Mbps. Also, SDSL connections are usually limited to a maximum distance of 12,000 feet between a DSL modem and its DSLAM.

- **Very High-Bit DSL (VDSL):** VDSL boasts a much higher bandwidth capacity than ADSL or SDSL, with a common downstream limit of 52Mbps and a limit of 12Mbps for upstream traffic. VDSL's distance limitation is 4000 feet of telephone cable between a cable modem and a DSLAM. This constraint might seem too stringent for many potential VDSL subscribers, based on their proximity to their closest telephone central office (CO). However, service providers and telephone companies offering VDSL service often extend their fiber optic network into their surrounding communities. This allows VDSL gateways to be located in multiple communities. The 4000 feet limitation then becomes a distance limitation between a DSL modem and the nearest VDSL gateway, thus increasing the number of potential VDSL subscribers.

Cable Modem

Cable television companies have a well-established and wide-reaching infrastructure for television programming. This infrastructure might contain both coaxial and fiber-optic cabling. Such an infrastructure is called a *hybrid-fiber coax (HFC)* distribution network. These networks can designate specific frequency ranges for upstream and downstream data transmission. The device located in a residence (or a business) that can receive and transmit in those data frequency ranges is known as a *cable* modem, as illustrated in Figure 2-22.

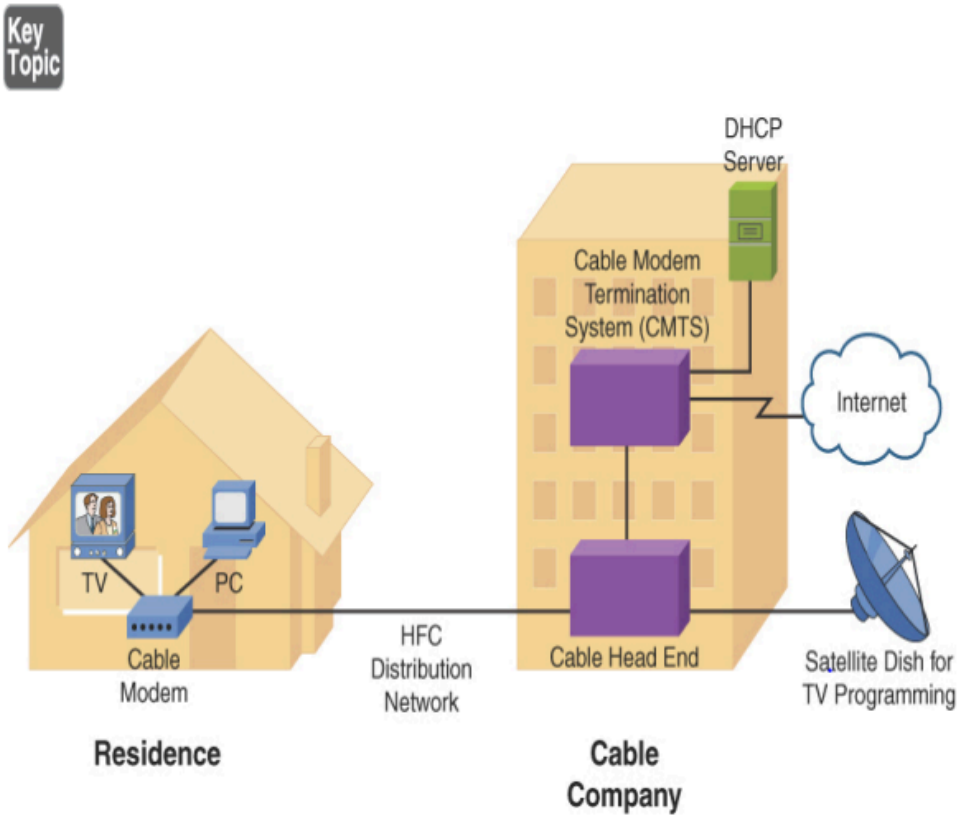


FIGURE 2-22 Cable Modem Sample Topology

The frequency ranges typically given for upstream and downstream data are as follows:

- **Upstream data frequencies:** 5MHz to 42MHz
- **Downstream data frequencies:** 50MHz to 860MHz

Although the theoretical maximum upstream/downstream bandwidth limits are greater (and are dependent on the HFC distribution network in use), most upstream speeds are limited to around 20Mbps, with downstream speeds limited to around 200Mbps. As HFC distribution networks continue to evolve, greater bandwidth capacities will be available. The current theoretical maximums are 1Gbps upstream and 10Gbps downstream. In my SOHO, for example, the main WAN connection using cable modem technology offers 300Mbps down and 100Mbps up. Of course, I pay extra for these faster speeds. As I write this, my service provider is trying to convince me to upgrade (and pay more) for 400Mbps download speeds and 400Mbps upload speeds.

The frequencies dedicated to data transmission are specified by a Data Over Cable Service Interface Specification (DOCSIS) version. Although DOCSIS is an international standard, European countries use their own set of frequency ranges, known as *Euro-DOCSIS*.

Leased Line

Often referred to as a dedicated leased line, a *leased line* is a point-to-point connection interconnecting two sites. All the bandwidth on that dedicated leased line is available to those sites. This means that, unlike with a packet-switched connection, the bandwidth of a dedicated leased line connection does not need to be shared among multiple service provider customers. WAN technologies used with dedicated leased lines include digital circuits, such as T1, E1, T3, and E3. These circuits can use multiplexing technology to simultaneously carry multiple conversations in different 64Kbps channels. A single 64Kbps channel is called a *Digital Signal 0 (DS0)*. When one of these circuits comes into your location, it terminates on a device called a channel service unit/data service unit (CSU/DSU). Also, be aware that a common Layer 2 protocol used on dedicated leased lines is Point-toPoint Protocol (PPP). A common connection type used to connect to a CSU/DSU is an RJ-48C, which looks like an RJ-45(Ethernet) connector.

Note:

High-Level Data Link Control (HDLC) is another protocol used on dedicated leased lines, though it is less common than PPP. HDLC lacks many of the features of PPP, and in its standards-based implementation, it can only support a single Layer 3 protocol on a circuit. However, Cisco has its own HDLC implementation in which the HDLC header has a protocol field, thus allowing the simultaneous transmission of multiple Layer 3 protocols.

T1

T1 circuits were originally used in telephony networks, with the intent of one voice conversation being carried in a single channel (that is, a single DS0). A T1 circuit is composed of 24 DS0s, which is called a *Digital Signal 1 (DS1)*. The bandwidth of a T1 circuit is 1.544Mbps:

- The size of a T1 frame = 193 bits (that is, 24 channels × 8 bits per channel + 1 framing bit = 193 bits).
- The *Nyquist theorem* needs 8000 samples to be sent per second for a voice conversation (that is, a rate at least twice the highest frequency of 4000Hz).
- Total bandwidth = 193-bit frames × 8000 samples per second = 1.544Mbps.

In a T1 environment, more than one frame is sent at once. Here are two popular approaches to grouping these frames:

- **Super Frame:** Combines 12 standard 193-bit frames into a *super frame*.
- **Extended Super Frame:** Combines 24 standard 193-bit frames into an *extended super frame.*

T1 circuits are popular in North America and Japan.

E1

An E1 circuit contains 32 channels, in contrast to the 24 channels on a T1 circuit. Only 30 of those 32 channels, however, can transmit data (or voice or video). Specifically, the first of those 32 channels is reserved for framing and synchronization, and the seventeenth channel is reserved for signaling (that is, setting up, maintaining, and tearing down a call). Because an E1 circuit has more DS0s than a T1, it has a higher bandwidth capacity. Specifically, an E1 has a bandwidth capacity of 2.048Mbps (8000 samples per second, as required by the Nyquist theorem, and 8 bits per sample × 32 channels = 2,048,000 bits per second).

Unlike a T1 circuit, an E1 circuit does not group frames in an SF or an ESF. Rather, an E1 circuit groups 16 frames in a *multiframe*. E1 circuits are popular outside North America and Japan.

T3

In the same T-carrier family of standards as T1, a T3 circuit offers an increased bandwidth capacity. Whereas a T1 circuit combines 24 DS0s into a single physical connection to offer 1.544Mbps of bandwidth, a T3 circuit combines 672 DS0s into a single physical connection, delivered to the customer over coaxial cable, which is called a *Digital Signal 3 (DS3).* T3 circuit has a bandwidth capacity of 44.7Mbps.

E3

Just as a T3 circuit provides more bandwidth than a T1 circuit, an E3 circuit's available bandwidth of 34.4Mbps is significantly more than the 2.048Mbps of bandwidth offered by an E1 circuit. A common misconception is that the bandwidth of E3 is greater than the bandwidth of T3 because E1's bandwidth is greater than T1's bandwidth. However, that is not the case; a T3 has greater bandwidth (that is, 44.7Mbps) than an E3 (that is, 34.4Mbps).

Metro-optical

Telecommunications networks today are often divided into a three-tier hierarchy, consisting of the access section, the metropolitan section, and the long-haul section. This metropolitan section clearly plays an important role and connects to both the access section and the long-haul section. Typically, this portion of the telecommunication network covers an area of 10 to 100 km and is often based on a SONET ring architecture. While this section is described using many different terms, CompTIA (and your author) like the term metro-optical for the technology used in this part of the WAN.

Synchronous Optical Network

Synchronous Optical Network (SONET) is a Layer 1 technology that uses fiber-optic cabling as its media. Because SONET is a Layer 1 technology, it can be used to transport various Layer 2 encapsulation types, such as ATM. Also, because SONET uses fiber-optic cabling, it offers high

data rates, typically in the 155Mbps to 10Gbps range, and long-distance limitations, typically in the 20 km to 250 km range. Optical Carrier transmission rates, such as OC3 (close to 155Mbps) and OC12 (close to 622Mbps), are examples of specifications for digital signal transmission bandwidth.

Note

The term *SONET* is often used synonymously with the term *Synchronous Digital Hierarchy (SDH)*, which is another fiber-optic multiplexing standard. Although these standards are similar, SONET is usually seen in North America, whereas SDH has greater worldwide popularity.

SONET networks can vary in their physical topology. For example, a SONET network can connect as many as 16 other devices in a linear fashion (similar to a bus topology) or in a ring topology. A metropolitan area network (MAN), as depicted in Figure 2-23, often uses a ring topology. The ring might circumnavigate a large metropolitan area. A site within that MAN could then connect to the nearest point on the SONET ring.

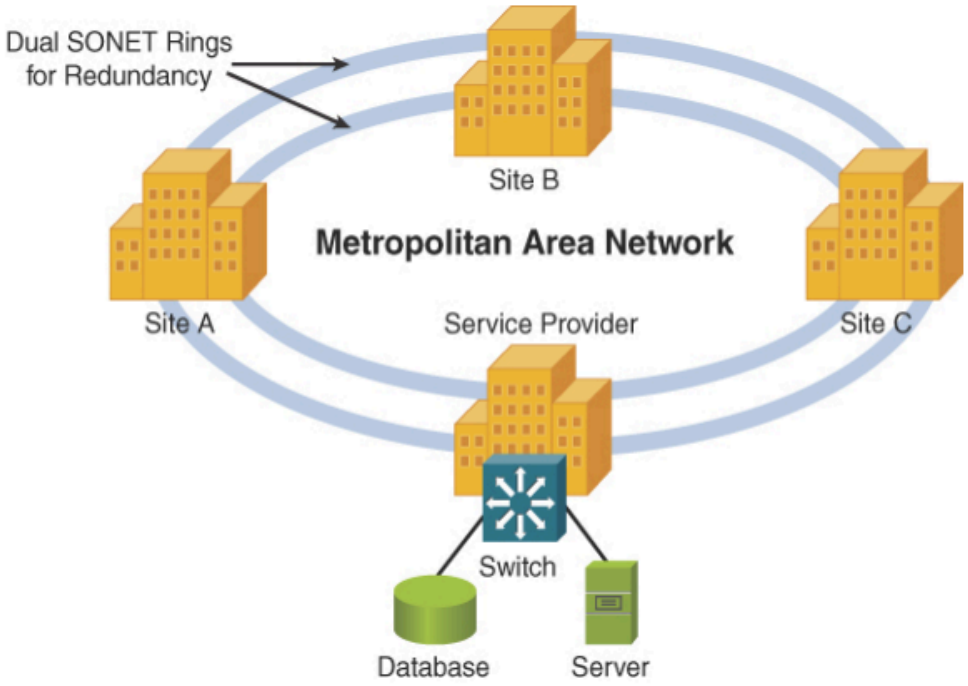


FIGURE 2-23 SONET Sample Topology

Note

A SONET network uses a single wavelength of light, along with time-division multiplexing (TDM), to support multiple data flows on a single fiber. This approach differs from dense wavelength-division multiplexing (DWDM), which is another high-speed optical network technology commonly used in MANs. DWDM uses as many as 32 light wavelengths on a single

fiber, where each wavelength can support as many as 160 simultaneous transmissions using more than 8 active wavelengths per fiber. Coarse wavelength-division multiplexing (CWDM) uses fewer than 8 wavelengths per fiber.

Note

Another optical WAN technology to be aware of is passive optical network (PON), which allows a single fiber cable to service as many as 128 subscribers. This is made possible via unpowered (that is, passive) optical splitters.

Real-World Case Study

The headquarters for Acme, Inc. is located on a single floor of a downtown building. Acme also has two branch offices, Branch1 and Branch2, that are in remote locations. The company wants to do file sharing, instant messaging, email, and voice on its own private networks when possible. It also wants connectivity to the Internet. At the headquarters location, Acme sets up a LAN with UTP (Cat 6a) cabling, with the clients and servers connected to a central switch.

This forms a physical star topology. For connectivity between HQ and its two branch offices, the company uses a service provider (SP) for WAN connectivity. The SP provides logical, point-to-point connections between the headquarters office and both of the branch locations. Physically, the path between the headquarters and each branch office is going through several routers in the SP's network. For the time being, Branch1 and Branch2 do not have direct connectivity to each other, so branch-to-branch traffic must pass through the headquarters site (hub and spoke). Next year, as more funds are available, the company can add WAN connectivity directly between Branch1 and Branch2. This will change the WAN topology from hub and spoke to full mesh.

Summary

Here are the main topics covered in this chapter:

- One way to classify networks is by their geographic dispersion. Specifically, these network types are identified in this chapter: LAN, WAN, CAN, MAN, PAN, WLAN, SAN, and SD-WAN.
- Another approach to classifying networks is based on a network's topology. Examples of network types, based on topology, include bus, ring, star, hybrid (partial mesh), full mesh, and hub and spoke. This chapter also provides information on the various wireless topologies available.
- This chapter contrasts client/server and peer-to-peer networks
- Multiprotocol label switching (MPLS) and multipoint generic routing encapsulation (mGRE) are detailed.
- Service-related entry points such as demarcation point and smartjack are covered.

- This chapter presents virtual networking components, including virtual network interface card (vNIC), virtual switch (vSwitch), network function virtualization (NFV), and hypervisor.
- This chapter covers provider links, including satellite, DSL, cable, leased line, and metro-optical.

Table 2-10 Key Topics for Chapter 2

Key Topic Element	Description	Page Number
List	Network types, as defined by geography	37

Tec

Key Topic Element	Description	Page Number
Figure 2-3	Service provider MPLS cloud	40
Table 2-2	Characteristics, benefits, and drawbacks of a client/server network	43
Table 2-3	Characteristics, benefits, and drawbacks of a peer-to-peer network	44
Table 2-4	Characteristics, benefits, and drawbacks of a bus topology	47
Table 2-5	Characteristics, benefits, and drawbacks of a ring topology	49
Table 2-6	Characteristics, benefits, and drawbacks of a star topology	50
Table 2-7	Characteristics, benefits, and drawbacks of a hub-and-spoke topology	51

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables”, or at least the section from this chapter and complete as many of the tables as possible from memory. Appendix D, “Memory Tables Answer Key”, includes the completed tables and lists so you can check your work.

Define Key Terms

Define the following key terms from this chapter and check your answers in the Glossary:

local area network (LAN) wide area network (WAN) campus area network (CAN) metropolitan area network (MAN) personal area network (PAN) wireless local area network (WLAN) storage area network (SAN) logical topology physical topology bus topology ring topology star topology hub-and-spoke topology full-mesh topology partial-mesh topology hybrid client/server network peer-to-peer network demarcation point smartjack network function virtualization (NFV) hypervisor satellite digital subscriber line (DSL) cable leased line

metro-optical

software-defined wide area network (SD-WAN) Multiprotocol Label Switching (MPLS) multipoint generic routing encapsulation (mGRE) vSwitch virtual network interface card (vNIC)

Additional Resources

Wide Area Network (WAN) technologies:

https://www.youtube.com/watch?v=9WkZT0YMZ70

MPLS basics: https://www.youtube.com/watch?v=mMu4iPWI1m8

Review Questions

The answers to these review questions appear in Appendix A, “Answers to Review Questions”.

1. What DSL variant has a distance limitation of 18,000 feet between a DSL modem and its DSLAM? a. HDSL b. ADSL c. SDSL d. VDSL
2. What kind of network do many cable companies use to service their cable modems with both fiber-optic and coaxial cabling? a. Head-end b. DOCSIS c. Composite d. HFC
3. Which of these components is used to make virtualization possible in the server environment by permitting multiple systems to use the underlying hardware of the host system? |||||||||||||||||||||||||||||||||||||||||||||||| a. vNIC b. vSwitch c. SD-WAN d. Hypervisor
4. Which technology allows enterprises to leverage a combination of transport services such as MPLS, 5G, LTE, or broadband to securely connect users to applications? a. mGRE b. SD-WAN c. Smartjack d. Demarcation
5. A company has various locations in a city interconnected using Metro Ethernet connections. This is an example of what type of network? a. WAN b. CAN c. PAN d. MAN
6. A network formed by interconnecting a PC to a digital camera via a USB cable is considered what type of network? a. WAN b. CAN c. PAN d. MAN
7. Which of the following physical LAN topologies requires the most cabling? a. Bus b. Ring Technet24 |||||||||||||||||||||||||||||||||||||||||||||||| c. Star d. WLAN
8. Which of the following topologies offers the highest level of redundancy? a. Full mesh b. Hub and spoke c. Bus d. Partial mesh
9. How many WAN links are required to create a full mesh of connections between five remote sites? a. 5 b. 10 c. 15 d. 20
10. Which of the following are advantages of a hub-and-spoke WAN topology as compared to a full-mesh WAN topology? (Choose two.) a. Lower cost b. Optimal routes c. More scalable d. More redundancy
11. Which type of network is based on network clients sharing resources with one another? a. Client/server b. Client/peer c. Peer-to-peer d. Peer-to-server
12. Which of the following is an advantage of a peer-to-peer network as compared with a client/server network? a. More scalable b. Less expensive c. Better performance d. Simplified administration
13. What network type would help facilitate communications when large video or audio files need to be housed and transferred through the network? a. WLAN b. CAN c. PAN d. SAN

Chapter 3: Network Media Types

This chapter covers the following topics related to Objective 1.3 (Summarize the types of cables and connectors and explain which is the appropriate type for a solution) of the CompTIA Network+ N10-008 certification exam:

- Copper
    - Twisted pair
        - Cat 5
        - Cat 5e
        - Cat 6
        - Cat 6a
        - Cat 7
        - Cat 8
    - Coaxial/RG-6
    - Twinaxial
    - Termination standards
        - TIA/EIA-568A
        - TIA/EIA-568B
    - Fiber
        - Single-mode
        - Multimode
    - Connector types
        - Local connector (LC), straight tip (ST), subscriber connector (SC), mechanical transfer (MT), registered jack (RJ)
            - Angled physical contact (APC)
            - Ultra-physical contact (UPC)
    - RJ11
    - RJ45
    - F-type connector
    - Transceivers/media converters
    - Transceiver type
        - Small form-factor pluggable (SFP)
        - Enhanced form-factor pluggable (SFP+)
        - Quad small form-factor pluggable (QSFP)
        - Enhanced quad small form-factor pluggable (QSFP+)
    - Cable management
        - Patch panel/patch bay
        - Fiber distribution panel
    - Punchdown block
        - 66
        - 110
        - Krone
        - Bix
    - Ethernet standards
        - Copper
            - 10BASE-T
            - 100BASE-TX
            - 1000BASE-T
            - 10GBASE-T

        - 40GBASE-T
    - Fiber
        - 100BASE-FX
        - 100BASE-SX
        - 1000BASE-SX
        - 10GBASE-SR
        - 10GBASE-LR
        - Coarse wavelength division multiplexing (CWDM)
        - Dense wavelength division multiplexing (DWDM)
        - Bidirectional wavelength division multiplexing (WDM)

Many modern networks have a daunting number of devices, and it is your job to understand the function of each device and how it works with the others. To create a network, these devices need some sort of interconnection. An interconnection uses one of a variety of media types. This chapter dives into the world of physical media. You will learn about classic media technologies that set the stage for the modern, high-speed media used in networks today. You will also learn about the connectors used for this media and the main aspects of cable management.

Foundation Topics: Copper and Fiber Media and Connectors

A network is an interconnection of devices. Those interconnections occur over some type of media. The media might be physical, such as a copper or fiber-optic cable. Alternatively, the media might be the air, through which radio waves propagate (as is the case with wireless networking technologies). |||||||||||||||||||||||||||||||||||||||||||| This section examines copper and fiber physical media types and the connectors they commonly use.

Coaxial Cable

Coaxial cable (referred to as *coax*) consists of two conductors. As illustrated in Figure 3-1, one of the conductors is an inner insulated conductor. This inner conductor is surrounded by another conductor that is sometimes made of a metallic foil or woven wire.

FIGURE 3-1 Coaxial Cable

Because the inner conductor is shielded by the metallic outer conductor, coaxial cable is resistant to electromagnetic interference (EMI). For example, EMI occurs when an external signal is received on a wire and might result in a corrupted data transmission. As another

example, EMI occurs when a wire acts as an antenna and radiates electromagnetic waves, which might interfere with data transmission on another cable. Coaxial cables have an associated characteristic impedance that needs to be balanced with the device (or terminator) with which the cable connects.

Note

The term *electromagnetic interference (EMI)* is sometimes used interchangeably with the term *radio frequency interference (RFI)*.

There are three common types of coaxial cables:

- **RG-59:** Typically used for short-distance applications, such as carrying composite video between two nearby devices. This cable type has loss characteristics such that it is not right for long-distance applications. RG-59 cable has a characteristic impedance of 75 ohms.
- ***RG-6:*** Used by local cable companies to connect individual homes to the cable company's distribution network. Like RG-59 cable, RG-6 cable has a characteristic impedance of 75 ohms.
- **RG-58:** Has loss characteristics and distance limitations like those of RG-59. However, the characteristic impedance of RG-58 is 50 ohms, and this type of coax was popular with early 10BASE2 Ethernet networks.

Although RG-58 coaxial cable was commonplace in early computer networks (that is, 10BASE2 networks), coaxial cable's role in modern computer networks is as the media used by cable modems. Cable modems are commonly installed in residences to provide high-speed Internet access over the same connection used to receive multiple television stations.

Note

Far less popular is ***twinaxial*** cabling, commonly called *twinax*. This is very similar to coaxial cable, but it uses two inner conductors instead of one.

Common connectors used on coaxial cables include the following:

<key_topic>

- **BNC:** A Bayonet Neill-Concelman (BNC) connector *(British Naval Connector)* in some literature can be used for a variety of applications, including as a connector in a 10BASE2 Ethernet network. A BNC coupler could be used to connect two coaxial cables together back-to-back.
- **F-connector:** An F-connector is often used for cable TV (including cable modem) connections. Notice that some, including CompTIA, refer to it simply as ***F-type connector.***

Figure 3-2 shows what both of these connectors look like.

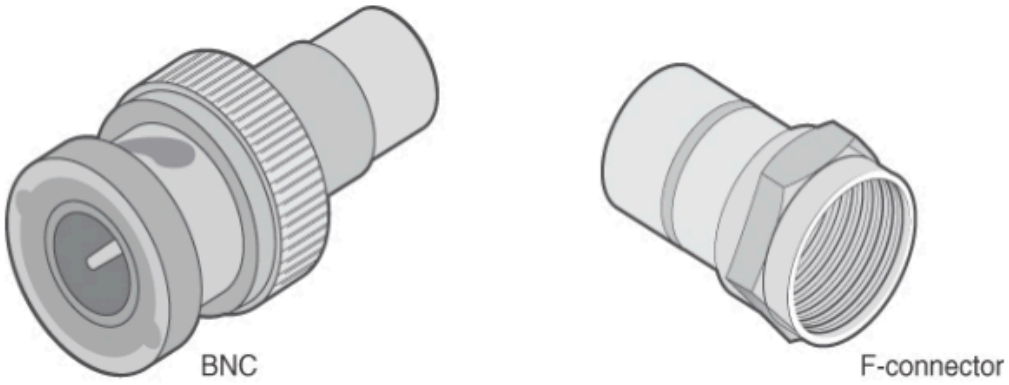


FIGURE 3-2 Coaxial Cable Connectors

Twisted-Pair Cable

Today's most popular LAN media type is *twisted-pair cable,* in which individually insulated copper strands are intertwined. Two categories of twisted-pair cable are shielded twisted pair (STP) and unshielded twisted pair (UTP). A UTP coupler could be used to connect two UTP cables, back-to-back. Also, for adherence to fire codes, you might need to select plenum-rated cable versus nonplenum cable.

To define industry-standard pinouts and color coding for twisted-pair cabling, the TIA/EIA-568 standard was developed. The first iteration of the TIA/EIA-568 standard, which was released in 1991, has come to be known as the ***TIA/EIA-568A*** standard.

Note

The TIA/EIA acronym comes from Telecommunications Industry Association/Electronic Industries Alliance.

In 2001, an updated standard was released, which became known as ***TIA/EIA-568B.*** Interestingly, the pinout of the two standards is the same. However, the color coding of the wiring is different. 568B is the more commonly used standard in the United States.

Shielded Twisted Pair

If wires in a cable are not shielded or twisted, the cable can act as an antenna, which might receive or transmit EMI. To help prevent this type of behavior, the wires (which are individually insulated) can be twisted together in pairs. If the distance between the twists is less than a

quarter of the wavelength of an electromagnetic waveform, the twisted pair of wires will not radiate that wavelength or receive EMI from that wavelength (in theory, if the wires were perfect conductors). However, as frequencies increase, wavelengths decrease.

One option for supporting higher frequencies is to surround a twisted pair in a metallic shielding, similar to the outer conductor in a coaxial cable.

One option for supporting higher frequencies is to surround a twisted pair in a metallic shielding, similar to the outer conductor in a coaxial cable. This type of cable is referred to as a *shielded-twisted pair (STP) cable.* Figure 3-3 shows an example of STP cable. The outer conductors shield the copper strands from EMI; however, the drawback of STP is that the addition of the metallic shielding adds to the expense of the cable.

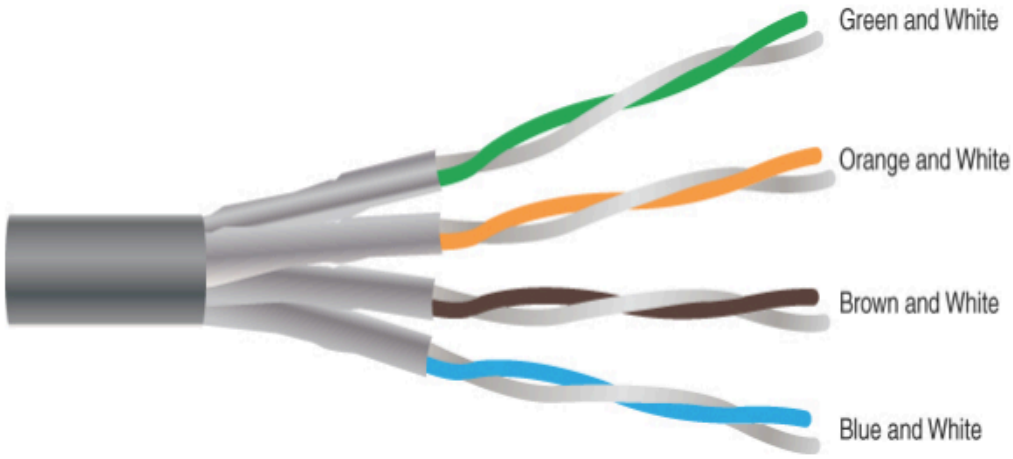


FIGURE 3-3 Shielded Twisted Pair

Unshielded Twisted Pair

Another way to block EMI from the copper strands making up a twisted pair cable is to twist the strands more tightly (that is, more twists per centimeter). With the strands tightly wrapped around each other, the wires insulate each other from EMI. Figure 3-4 illustrates an example of UTP cable. Because UTP is less expensive than STP, it has grown in popularity since the mid-1990s to become the media of choice for most LANs.

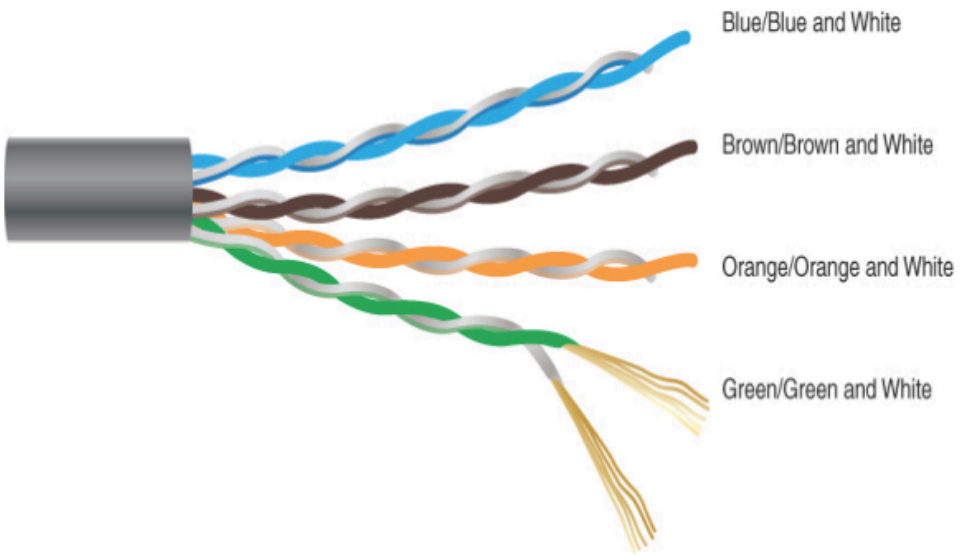


FIGURE 3-4 Unshielded Twisted Pair

FIGURE 3-4 Unshielded Twisted Pair

UTP cable types vary in their data-carrying capacity. Common categories of UTP cabling include the following:

<key_topic>

- **Category 5:** Category 5 *(Cat 5)* cable is commonly used in Ethernet 100BASE-TX networks, which carry data at a rate of 100Mbps. However, Cat 5 cable can carry ATM traffic at a rate of 155Mbps. Most Cat 5 cables consist of four pairs of 24-gauge wires. Each pair is twisted, with a different number of twists per meter. However, on average, one pair of wires has a twist every 5 cm.
- **Category 5e:** Category 5e *(Cat 5e)* cable is an updated version of Cat 5 that is commonly used for 1000BASE-T networks, which carry data at a rate of 1Gbps. Cat 5e cable offers reduced crosstalk compared to Cat 5 cable.
- **Category 6:** Like Cat 5e cable, Category 6 (*Cat 6)* cable is commonly used for 1000BASE-T Ethernet networks. Some Cat 6 cable is made of thicker conductors (for example, 22-gauge or 23-gauge wire), although some Cat 6 cable is made from the same 24-gauge wire used by Cat 5 and Cat 5e. Cat 6 cable has thicker insulation and offers reduced crosstalk compared with Category 5e.
- **Category 6a:** Category 6a *(Cat 6a)*, or augmented Cat 6, supports twice as many frequencies as Cat 6 and can be used for 10GBASE-T networks, which can transmit data at a rate of 10 billion bits per second (10Gbps).
- **Category 7:** ***Cat 7*** is not an IEEE standard, and it is not very popular as a result. This very strict specification supports 10Gbps over 100 m using copper media.

- **Category 8:** ***Cat 8*** is capable of 40Gbps speeds. As you might guess, however, this speed comes at a cost. Cat 8 supports distances of only 30 to 36 m, depending on the patch cables used. These short distances and very high speeds are ideal for connections in a data center between high-speed multilayer switches.

Although other wiring categories exist, the ones presented in this list are the categories most commonly seen in modern networks. Most UTP cabling used in today's networks is considered to be *straight-through*, meaning that the RJ45 jacks at each end of a cable have matching pinouts. For example, pin 1 in an RJ45 jack at one end of a cable uses the same copper conductor as pin 1 in the RJ45 jack at the other end of a cable. However, some network devices cannot be interconnected with a straightthrough cable. For example, consider two PCs interconnected with a straight-through cable. Because the network interface cards (NICs) in these PCs use the same pair of wires for transmission and reception, when one PC sends data to the other PC, the receiving PC would receive the data on its transmission wires rather than its reception wires. For such a scenario, you |||||||||||||||||||||||||||||||||||||||||||||||| can use a crossover cable, which swaps the transmit and receive wire pairs between the two ends of a cable.

Note

A crossover cable for Ethernet devices is different from a crossover cable used for a digital T1 circuit. Specifically, an Ethernet crossover cable has a pin mapping of 1 → 3, 2 → 6, 3 → 1, and 6 → 2, whereas a T1 crossover cable has a pin mapping of 1 → 4, 2 → 5, 4 → 1, and 5 → 2. Another type of cable is the rollover cable, which is used to connect to a console port to manage a device such as a router or switch. The pin mapping for a rollover cable is 1 ↔ 8, 2 ↔ 7, 3 ↔ 6, 4 ↔ 5. The end of the cable looks like an RJ45 eight-pin connector.

Note

A traditional port found in a PC's NIC is called a *media-dependent interface (MDI).* If a straight-through cable connects a PC's MDI port to an Ethernet switch port, the Ethernet switch port needs to swap the transmit pair of wires (that is, the wires connected to pins 1 and 2) with the receive pair of wires (that is, the wires connected to pins 3 and 6). Therefore, a traditional port found on an Ethernet switch is called a *media-dependent interface crossover (MDIX),* and it reverses the transmit and receive pairs. However, if you want to interconnect two switches, where both switch ports used for the interconnection are MDIX ports, the cable needs to be a crossover cable. Fortunately, most modern Ethernet switches have ports that can automatically detect whether they need to act as MDI ports or MDIX ports and make the appropriate adjustments. This eliminates the necessity of using straight-through cables for some Ethernet switch connections and crossover cables for other connections. With this *Auto-MDIX* feature, you can use either straight-through cables or crossover cables.

Twisted-Pair Cable Connectors

Common connectors used on twisted-pair cables are as follows:

<key_topic>

RJ45: A type 45 registered jack (RJ45) is an eight-pin connector found in most Ethernet networks. However, most Ethernet implementations only use four of the eight pins.

RJ11: A type 11 registered jack (RJ11) has the capacity to be a six-pin connector. However, most RJ11 connectors have only two or four conductors. An RJ11 connector is found in most home telephone networks. However, most home phones use only two of the six pins.

DB-9 (RS-232): A nine-pin D-subminiature (DB-9) connector is an older connector used for low-speed asynchronous serial communications, such as a PC to a serial printer, a PC to a console part of a router or switch, or a PC to an external modem. Do not confuse the DB-9 with the DB-25. The DB-25 connector was also used for the serial or parallel ports of early computers.

Figure 3-5 shows what these connectors look like.

FIGURE 3-5 Twisted-Pair Cable Connectors

Plenum versus Nonplenum Cable

If a twisted-pair cable is to be installed under raised flooring or in an openair return, fire codes must be considered. For example, imagine that there was a fire in a building. If the outer insulation of a twisted-pair cable caught on fire or started to melt, it could release toxic fumes. If those toxic fumes were released in a location such as an open-air return, those fumes could be spread throughout a building, posing a huge health risk.

To mitigate the concern of pumping poisonous gas throughout a building's heating, ventilation, and air conditioning (HVAC) system, *plenum* cabling can be used. The outer insulator of a plenum twisted-pair cable is not only fire retardant; in addition, some plenum cabling uses a fluorinated ethylene polymer (FEP) or a low-smoke polyvinyl chloride (PVC) to minimize dangerous fumes.

Note

Check with your local fire codes before installing network cabling.

Fiber-Optic Cable

An alternative to copper cabling is fiber-optic cabling, which sends light (instead of electricity) through an optical fiber (typically made of glass). Using light instead of electricity makes fiber optics immune to EMI. Also, depending on the Layer 1 technology being used, fiber-optic cables typically have greater range (that is, a greater maximum distance between networked devices) and greater data-carrying capacity. Lasers are often used to inject light pulses into a fiber-optic cable. However, lower-cost light-emitting diodes (LEDs) are also available. Fiber-optic cables are generally classified according to their diameter and fall into one of two categories: multimode fiber (MMF) and single-mode fiber (SMF). The wavelengths of light also vary between MMF and SMF cables. Usually, wavelengths of light in an MMF cable are in the range 850–1300 nm, where nm stands for nanometers. (A nanometer is one-billionth of a meter.) Conversely, the wavelengths of light in an SMF cable are usually in the range 1310–1550 nm. A fiber coupler could be used to connect two fiber cables, back-to-back.

Multimode Fiber

When a light source, such as a laser, sends light pulses into an fiber-optic cable, what keeps the light from simply passing through the glass and being dispersed into the surrounding air? The trick is that fiber-optic cables use two different types of glass. There is an inner strand of glass (that is, a core) surrounded by an outer cladding of glass, similar to the construction of the previously mentioned coaxial cable. The light injected by a laser (or LED) enters the core, and the light is prevented from leaving that inner strand and going into the outer cladding of glass. Specifically, the indexes of refraction of these two different types of glass are so different that if the light attempts to leave the inner strand, it hits the outer cladding and bends back on itself. To better understand this concept, consider a straw in a glass of water, as shown in Figure 3-6. Because air and water have different indexes of refraction (that is, light travels at a slightly different speeds in air and water), the light that bounces off the straw and travels to our eyes is bent by the water's index of refraction. When a fiber-optic cable is manufactured, dopants are injected into the two types of glasses, making up the core and cladding to give them significantly different indexes of refraction, thus causing any light attempting to escape to be bent back into the core. The path that light takes through a fiber-optic cable is called a mode of propagation. The diameter of the core in a multimode fiber is large enough to permit light to enter the core at different angles, as depicted in Figure 3- 7. If light enters at a steep angle, it bounces back and forth much more frequently on its way to the far end of the cable than does light that enters the cable perpendicularly. If pulses of light representing different bits travel down the cable using different modes of propagation, it is possible that the bits (that is, the pulses of light representing the bits) will arrive out of order at the far end (where the pulses of light, or absence of light, are interpreted as binary data by photoelectronic sensors).

FIGURE 3-6 Example: Refractive Index

<key_topic>

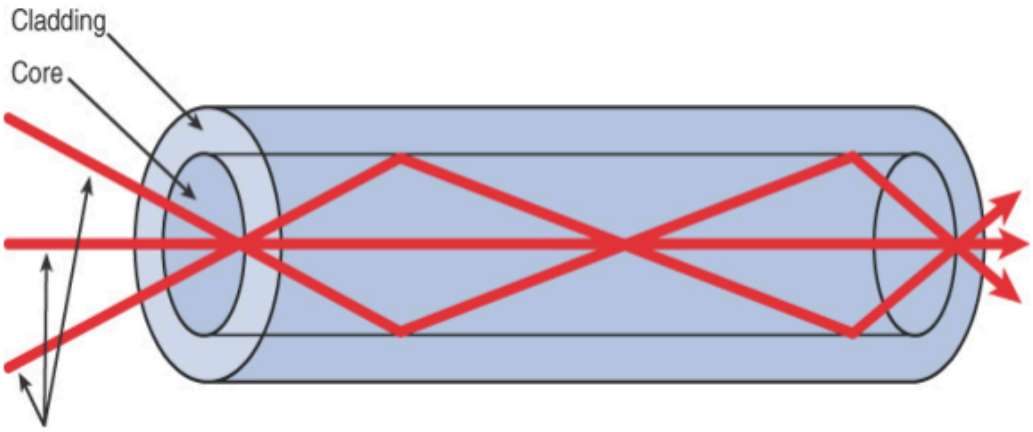


FIGURE 3-7 Light Propagation in Multimode Fiber

For example, say that the pulse of light representing the first bit intersects the core at a steep angle and bounces back and forth many times on its way to the far side end of the cable, while the light pulse representing the second bit intersects the core perpendicularly and does not bounce back and forth very much. With all of its bouncing, the first bit has to travel further than the second bit, and so the bits might arrive out of order. This condition is known as *multimode delay distortion.* To mitigate multimode delay distortion, *multimode fiber* (MMF) is typically limited to shorter distances than SMF.

Single-Mode Fiber

Single-mode fiber (SMF) eliminates the issue of multimode delay distortion by having a core with a diameter so small that it only permits one mode (that is, one path) of propagation, as shown in Figure 3-8. With the issue of multimode delay distortion mitigated, SMF typically can be run for longer distances than MMF.

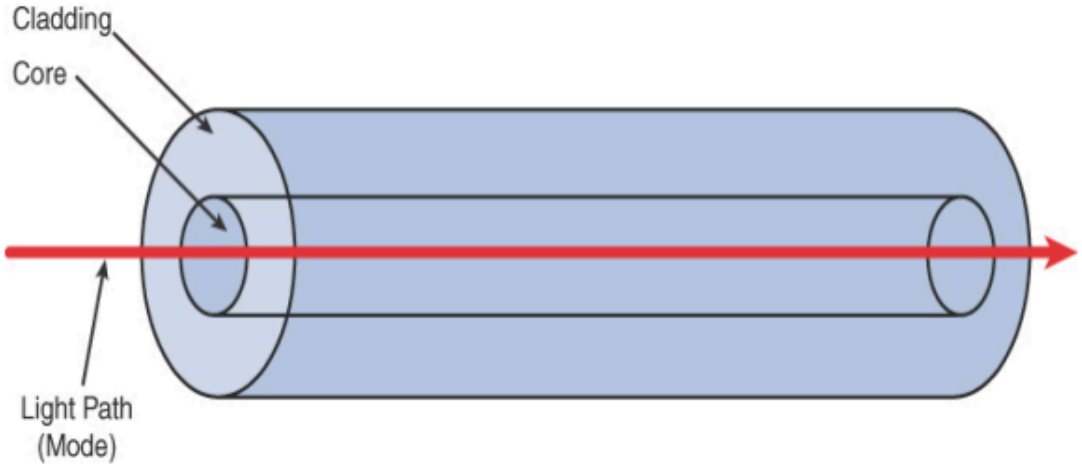


FIGURE 3-8 Light Propagation in Single-Mode Fiber

Key Topic

A potential downside to SMF, however, is cost. Because SMF has to be manufactured to very exacting tolerances, you usually pay more for a given length of single-mode fiber-optic cabling. However, for some implementations where greater distances are required, the cost is an acceptable trade-off for reaching greater distances.

Fiber-Optic Cable Connectors

Some common connectors used on fiber-optic cables are as follows:

- **ST:** A *straight tip (ST)* connector is sometimes referred to as a *bayonet connector*, because of the long tip extending from the connector. ST connectors are most commonly used with MMF. You connect an ST connector to a terminating device by pushing the connector into the terminating equipment and then twisting the connector housing to lock it in place.
- **SC:** Different literature defines an SC connector as a *subscriber connector, standard connector, or square connector.* You connect an SC connector by pushing it onto the terminating device; you can remove it by pulling the connector from the terminating device. This Technet24 |||||||||||||||||||||||||||||||||||||||| connector type has slight variants within the industry, with the major types being APC, UPC, and MTRJ. Always consult with the vendor or an IT staff member regarding the exact requirements.

- **LC:** You connect a *Lucent connector, little connector,* or *local connector* to a terminating device by pushing the connector into the terminating device. You can remove it by pressing the tab on the connector and pulling it out of the terminating device.
- **MTRJ:** The most unique characteristic of a ***media termination recommended jack (MTRJ)*** or ***mechanical transfer (MT) registered jack (RJ)*** connector is that two fiber strands (a transmit strand and a receive strand) are included in a single connector. You connect an MTRJ connector by pushing it into the terminating device; you can remove it by pulling the connector from the terminating device.

Figure 3-9 shows what these connectors look like.

Fiber Connector Polishing Styles

Fiber-optic cables have different types of mechanical connections. The type of connection impacts the quality of the fiber-optic transmission. Listed from basic to better, the options include physical contact (PC), *ultra physical contact (UPC), and angled physical contact (APC),* which refer to the polishing styles of fiber-optic connectors. The different polish of the fiber-optic connectors results in different performance of the connector. The less back reflection, the better the transmission. The PC back reflection is – 40 dB, the UPC back reflection is around –55 dB, and the APC back reflection is about –70 dB.

Ethernet and Fiber Standards

A popular implementation of Ethernet, in the early days, was called *10BASE5*. The 10 in 10BASE5 referred to network throughput, specifically 10Mbps (that is, 10 million [mega] bits per second). The BASE in 10BASE5 referred to baseband, as opposed to broadband. Finally, the 5 in 10BASE5 indicated the distance limitation of 500 m. The cable used in 10BASE5 networks, illustrated in Figure 3-10, was a larger diameter than most types of media. In fact, this network type became known as *thicknet*.

Key Topic

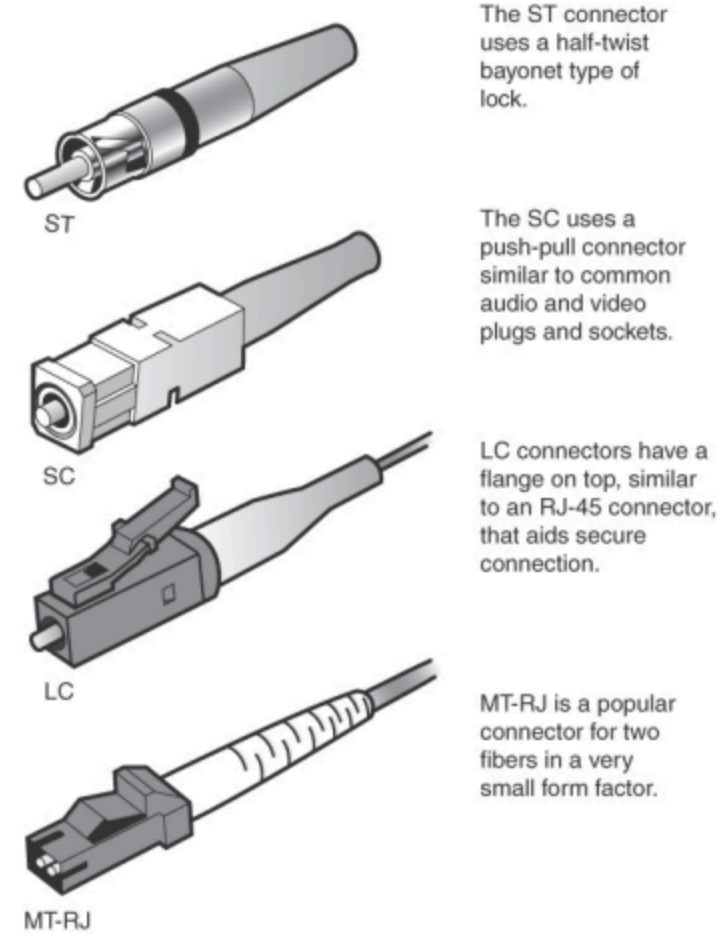


FIGURE 3-9 Common Fiber-Optic Connectors

Another early Ethernet implementation was 10BASE2. From the previous analysis of 10BASE5, you might conclude that 10BASE2 was a 10Mbps baseband technology with a distance limitation of 200 meters. That is almost correct. However, 10BASE2's actual distance limitation was 185 m. The cabling used in 10BASE2 networks was significantly thinner and therefore less expensive than 10BASE5 cabling. As a result, 10BASE2 cabling, illustrated in Figure 3-11, was known as *thinnet* or *cheapnet*.

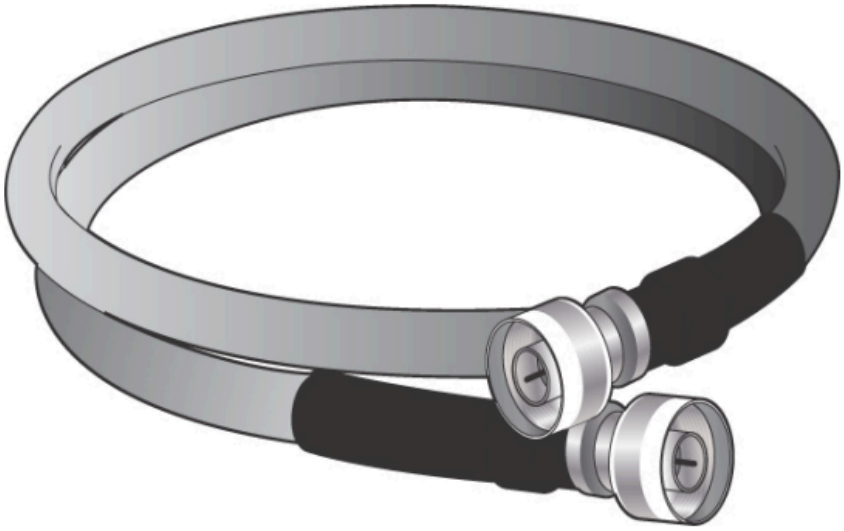


FIGURE 3-10 10BASE5 Cable

FIGURE 3-11 Coaxial Cable Used for 10BASE2

10BASE5 and 10BASE2 networks are rarely, if ever, seen today. The cabling used by these legacy technologies quickly faded in popularity with the advent of UTP cabling. The 10Mbps version of Ethernet that relied on UTP cabling, an example of which is provided in Figure 3-12, is known as *10BASE-T*. Notice that the “T” in 10BASE-T refers to twisted-pair cabling.

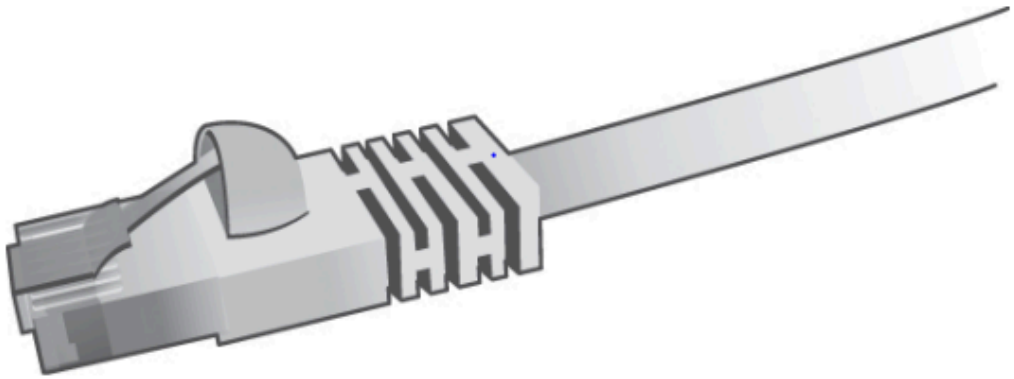


FIGURE 3-12 UTP Cable Used for 10BASE-T

Distance and Speed Limitations

To understand the bandwidth available on networks, you need to understand a few terms. You should already know that a *bit* refers to one of two values. These values are represented using binary math, which only uses the numbers 0 and 1. On a cable such as twisted-pair cable, a bit could be represented by the absence or presence of voltage. Fiber optic cables, however, might represent a bit with the absence or presence of light. The bandwidth of a network is measured in terms of how many bits the network can transmit during a 1-second period of time. For example, if a network has the capacity to send 10,000,000 (that is, 10 million) bits in a 1-second period of time, the bandwidth capacity is said to be 10 megabits (that is, millions of bits) per second (or *Mbps*). Table 3-1 defines common bandwidths supported on distinct types of Ethernet networks.

<key_topic>

Table 3-1 Ethernet Bandwidth Capacities

Ethernet Type	Bandwidth Capacity
Standard Ethernet	10Mbps: 10 million bits per second (that is, 10 megabits per second)
Fast Ethernet	100Mbps: 100 million bits per second (that is, 100 megabits per second)
Gigabit Ethernet	1Gbps: 1 billion bits per second (that is, 1 gigabit per second)
10-Gigabit Ethernet	10Gbps: 10 billion bits per second (that is, 10 gigabits per second)
100-Gigabit Ethernet	100Gbps: 100 billion bits per second (that is, 100 gigabits per second)

The type of cabling used in an Ethernet network influences the bandwidth and the distance limitation of the network. For example, fiber-optic cabling often has a higher bandwidth capacity and can be run longer distances than twisted-pair cabling. As mentioned earlier in this chapter, because of the issue of multimode delay distortion, SMF usually has a longer distance limitation than MMF. Although not comprehensive, Table 3-2 lists a number of Ethernet standards, along with the media type, bandwidth capacity, and distance limitation for each.

<key_topic>

Table 3-2 Types of Ethernet

Ethernet Standard	Media Type	Bandwidth Capacity	Distance Limitation
10BASE5	Coax (thicknet)	10Mbps	500 m
10BASE2	Coax (thinnet)	10Mbps	185 m
10BASE-T	Cat 3 (or higher) UTP	10Mbps	100 m
100BASE-TX	Cat 5 (or higher) UTP	100Mbps	100 m
100BASE-FX	MMF	100Mbps	2 km
100BASE-SX	MMF	100Mbps	850 nm
1000BASE-T	Cat 5e (or higher) UTP	1Gbps	100 m
1000BASE-TX	Cat 6 (or higher) UTP	1Gbps	100 m
1000BASE-SX	MMF	1Gbps	550 km
1000BASE-LX	SMF	1Gbps	5 km
1000BASE-LH	SMF	1Gbps	10 km

10GBASE-SR	MMF	10Gbps	26–400 m
10GBASE-LR	SMF	10Gbps	10–25 km
10GBASE-ER	SMF	10Gbps	40 km
10GBASE-SW	MMF	10Gbps	300 m
10GBASE-LW	SMF	10Gbps	10 km
10GBASE-EW	SMF	10Gbps	40 km
10GBASE-T	Cat 6a (or higher)	10Gbps	100 m
40GBASE-T	Cat 8	40Gbps	30 m
100GBASE-SR10	MMF	100Gbps	125 m
100GBASE-LR4	SMF	100Gbps	10 km
100GBASE-ER4	SMF	100Gbps	40 km

Note

Two often-confused terms are *100BASE-T* and *100BASE-TX*. 100BASE-T is not a specific standard. Rather, 100BASE-T is a category of standards and includes 100BASE-T2 (which uses two pairs of wires in a Cat 3 cable), 100BASE-T4 (which uses four pairs of wires in a Cat 3 cable), and 100BASE-TX. 100BASE-T2 and 100BASE-T4 were early implementations of 100Mbps Ethernet and are no longer used. Therefore, you can generally use the terms *100BASE-T* and *100BASE-TX* interchangeably. Similarly, the term *1000BASE-X* is not a specific standard. Rather, 1000BASE-X refers to all Ethernet technologies that transmit data at a rate of 1Gbps over fiber-optic cabling. Additional and creative ways of using Ethernet technology include IEEE 1901-2013, which could be used for Ethernet over HDMI cables and Ethernet over existing power lines to avoid having to run a separate cabling just for networking.

Transceivers

When you want to uplink one Ethernet switch to another, you might need different connectors (for example, for MMF, SMF, or UTP) for different installations. Fortunately, some Ethernet switches have one or more empty slots in which you can insert a gigabit interface converter (GBIC). GBICs are interfaces that have a bandwidth capacity of 1Gbps and are available with MMF, SMF, and UTP connectors. This allows you to have flexibility in the uplink technology you use in an Ethernet switch. A smaller version of a regular GBIC is the *small-form factor pluggable*

(SFP), which is sometimes called a *mini-GBIC*. And to show the variety of *transceivers* you might encounter today, even this SFP has many variations, including the following:

- *Enhanced form-factor pluggable (SFP+)*
- *Quad small form-factor pluggable (QSFP)*
- *Enhanced quad small-form pluggable (QSFP+)*

Multiplexing in Fiber-Optic Networks

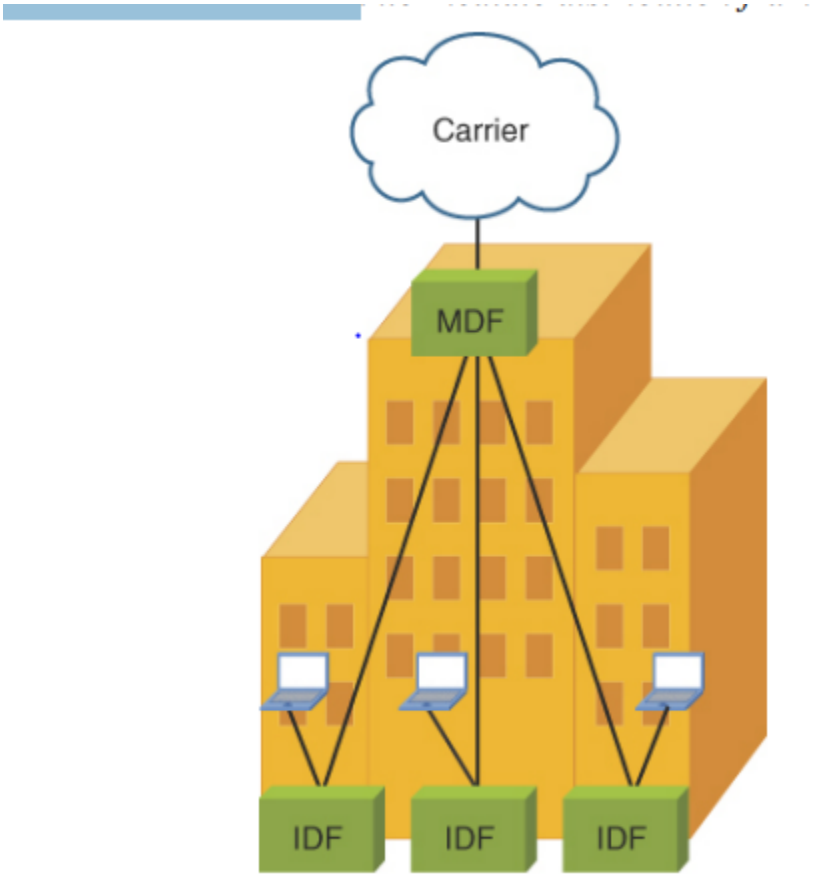
Remember, as mentioned in Chapters 1, "The OSI Model and Encapsulation," and 2, "Network Topologies and Types," that multiplexing allows multiple communications sessions to share the same physical medium. At this point, you need to be familiar with three different approaches common with fiber networking:

- ***Dense wavelength-division multiplexing (DWDM):*** DWDM uses as many as 32 light wavelengths on a single fiber, where each wavelength can support as many as 160 simultaneous transmissions using more than eight active wavelengths per fiber.
- ***Coarse wavelength-division multiplexing (CWDM):*** CWDM uses fewer than eight active wavelengths per fiber.
- ***Bidirectional wavelength-division multiplexing (WDM):*** This approach multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths. Using this technique enables bidirectional communications over one strand of fiber and increases the overall capacity.

Cable Management

After deciding what type of media you are going to use in your network (for example, UTP, STP, MMF, or SMF), you should install that media as part of an organized cable distribution system. Typically, cable distribution systems are hierarchical in nature. Consider the example profiled in Figure 3-13. In this example, cable from end-user offices runs back to common locations within the building. These locations are sometimes referred to as wiring closets. Cables in these locations might terminate in a *patch panel*, or a *patch bay*. The patch panel might consist of some sort of cross-connect block wired into a series of ports (for example, RJ45 ports), which can be used to quickly interconnect cables coming from end-user offices with a network device, such as an Ethernet switch. A common term for cross-connect blocks is *punchdown blocks*. This term describes physically how you connect the media – "punching" the media into the appropriate slot.

The fiber connections into a wiring closet can terminate into a *fiber distribution panel*, also known as a fiber-optic patch panel. This cable management system is mainly used for accommodating fiber panel terminations, connections, and patching. The two major categories of fiber distribution panels are wall mount and rack mount types. A building might have multiple patch panels (for example, on different floors of the building). Common locations where cables from nearby offices terminate are often called *intermediate distribution frames (IDFs)*.



GURE 3-13 Example: Cable Distribution System

Figure 3-13, Example: Cable Distribution System

<key_topic>

The two most popular types of cross-connect blocks found in an IDF are detailed here:

- ***66 block:*** 66 blocks were traditionally used in corporate environments for cross-connecting phone system cabling. As 10Mbps LANs grew in popularity, in the late 1980s and early 1990s, these termination blocks were used to cross-connect Cat 3 UTP cabling. The electrical characteristics (specifically, cross-talk) of a 66 block, however, do not support higher-speed LAN technologies, such as 100Mbps Ethernet networks. Figure 3-14 illustrates a 66 block.

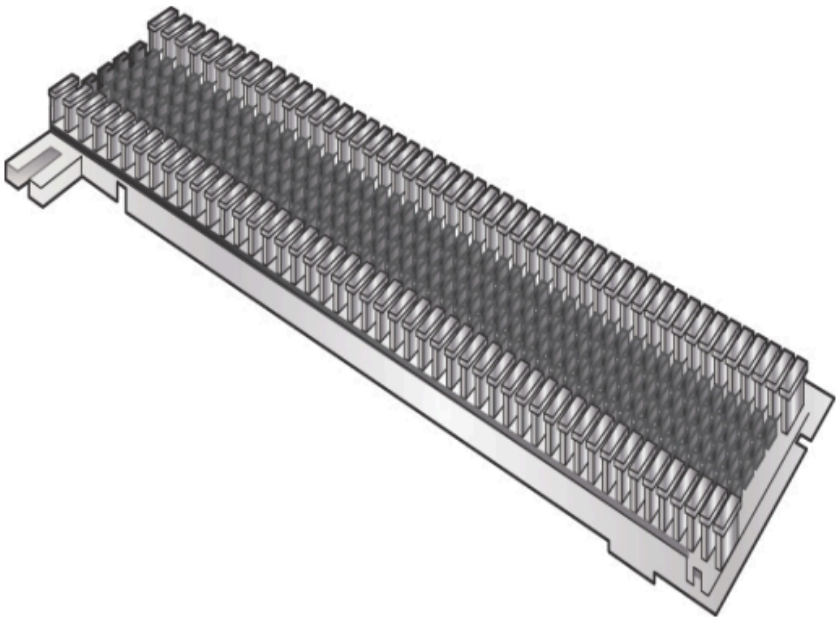


FIGURE 3-14 66 Block

- ***110 block:*** Because 66 blocks are subject to a lot of crosstalk (that is, interference between different pairs of wires) for higher-speed LAN connections, 110 blocks are often used to terminate cable (for example, a Cat 5e cable) used for higher-speed LANs. Figure 3-15 illustrates a 110 block.

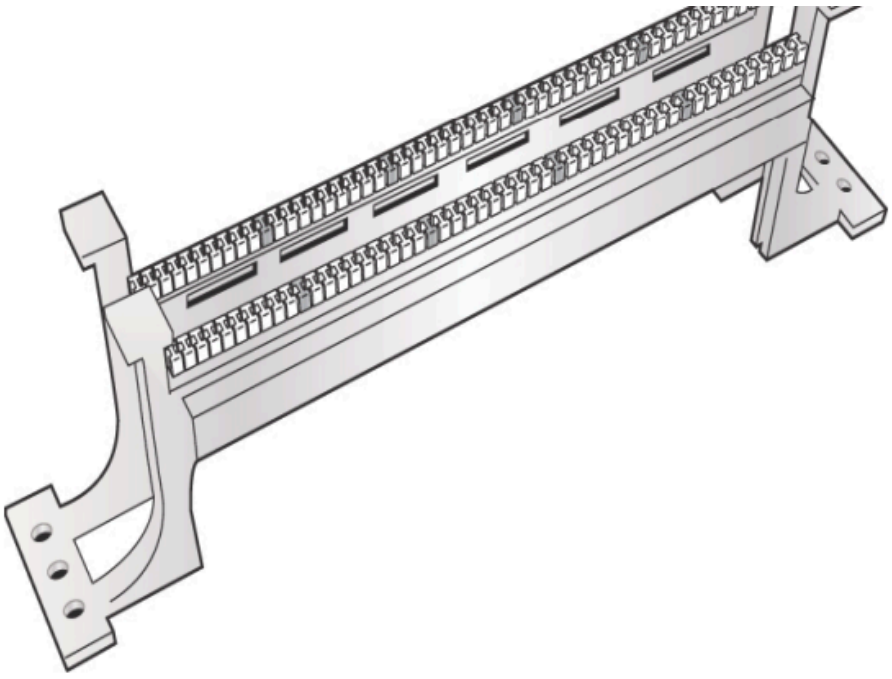


FIGURE 3-15 110 Block

There are two other cross-connect blocks you should be familiar with:

- *Krone* **(or Krone-LSA Plus):** Krone, a proprietary European alternative to 110 block, is used not only in data environments but in television broadcasting.
- *BIX* **(or Building Industry Cross-connect):** A BIX terminates 25 pairs (that is, 50 wires). The 25 pairs may be punched down to one side of a "wafer" that is then inserted into a metal frame with the punched down side against the wall, so you only see the unused side.

The centralized distribution frame, which connects out to multiple IDFs, is called the *main distribution frame (MDF).*

Note

You will learn more about IDFs and MDFs in Chapter 14, "Organizational Documents and Policies," which covers the importance of documenting these designs and implementations as part of operational excellence for your organization.

Media Converters

Due to the wide variety of copper and fiber cabling used by different network devices, you might need one or more media converters.

Examples of media converters include the following:

- MMF to Ethernet
- SMF to Adapter
- Fiber to coaxial
- SMF to MMF

Real-World Case Study

Acme, Inc. is analyzing their physical media implementations and the performance they are achieving in their Ethernet-based LAN. The decision has been made to retain portions of the Cat 5e-based media, which is capable of 1Gbps speeds. Other areas of the network, particularly those that must deal with aggregate bandwidth demands, are being upgraded to Cat 8. Remember, Category 8 is capable of 40Gbps speeds. Cat 8 supports distances of only 30 to 36 m, depending on the patch cables used. For Acme, these short distances and very high speeds are ideal for connections between high-speed multilayer switches that make up a good part of the distribution and core layers of the LAN.

Summary

Here are the main topics covered in this chapter:

- There are different options today when it comes to copper and fiber media and connectors. This section of the chapter explored many of these in great detail.
- This chapter also covered the different options for multiplexing in fiber-optic environments.
- Finally, this chapter examined some typical component types for cable management.

Exam Preparation Tasks

Review All the Key Topics

Review the most important topics from this chapter, noted with the Key Topic icon in the outer margin of the page. Table 3-3 lists these key topics and the page number where each is found.

<key_topic>

Table 3-3 Key Topics for Chapter 3

Key Topic Element	Description	Page Number
List, Figure 3-2	Common coaxial connectors	81
List	Categories of UTP cabling	83
List, Figure 3-5	Twisted-pair cable connectors	85
Figure 3-7	Light propagation in multimode fiber	88
Figure 3-8	Light propagation in single-mode fiber	89

Te

Key Topic Element	Description	Page Number
List, Figure 3-9	Common fiber-optic connectors	91
Table 3-1	Ethernet bandwidth capacities	93
Table 3-2	Types of Ethernet	94
Figure 3-13	Cable distribution system example	97

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables”, or at least the section for this chapter and complete as many of the tables as possible from memory. Appendix D, “Memory Tables Answer Key,” includes the completed tables and lists so you can check your work.

Define Key Terms

Define the following key terms from this chapter and check your answers in the Glossary:

Twisted pair
Cat 5
Cat 5e
Cat 6
Cat 6a
Cat 7
Cat 8
coaxial/RG-6
twinaxial
TIA/EIA-568A
TIA/EIA-568B
Single-mode fiber (SMF)
Multi-mode fiber (MMF)
Local connector (LC)
Straight-tip (ST) connector
Mechanical transfer (MT)
Registered jack (RJ)
40GBASE-T 100BASE-FX 100BASE-SX 1000BASE-SX 1000BASE-LX 10GBASE-SR 10GBASE-LR coarse wavelength-division multiplexing (CWDM) dense wavelength-division multiplexing (DWDM)angled physical contact (APC) ultra-physical contact (UPC) RJ11 RJ45 F-type connector transceiver media converter small form-factor pluggable (SFP) enhanced form-factor pluggable (SFP+) quad small form-factor pluggable (QSFP) enhanced quad small form-factor pluggable (QSFP+) patch panel/patch bay fiber distribution panel punchdown block 66 block 110 block Krone Bix 10BASE-T 100BASE-TX 1000BASE-T 10GBASE-T 40GBASE-T 100BASE-FX 100BASE-SX
Bidirectional wavelength-division multiplexing (WDM)

Additional Resources

Fundamentals of Fiber Optic Cabling: https://youtu.be/- VYhfR8Fv2I **Real World Cabling:** https://www.youtube.com/watch?v=CV3- is8Yd8U **Review Questions**

The answers to these review questions appear in Appendix A: “Answers to Review Questions”

1. Which of the following categories of UTP cabling are commonly used for 1000BASE-T networks? (Choose two.) a. Cat 5 b. Cat 5e c. Cat 6 d. Cat 6f 2. Which type of cable

might be required for installation in a drop ceiling that is used as an open-air return duct? a. Riser b. Plenum c. Multimode d. Twinaxial 3. Which of the following is the eight-pin connector found in most Ethernet networks? a. RJ11 b. RJ45 c. DB-9 ||||||||||||||||||||||||||||||||||||||||| d. ST 4. What are the two major categories of fiber-optic media? (Choose two.) a. Straight-through b. Single-mode c. Unshielded d. Multimode

5. What is the speed of Fast Ethernet? a. 1Mbps b. 100Mbps c. 1Gbps d. 10Gbps 6. What is a common fiber-optic cable management component for terminating many connections? a. A fiber distribution panel b. An IDF c. A 110 block d. A plenum 7. What Ethernet UTP cable was the first standard cable to carry data at speeds of 1Gbps? a. Cat 4 b. Cat 5 c. Cat 5e d. Cat 6

8. What type of connector is often used for cable TV (including cable modem) connections? a. Krone Technet24 ||||||||||||||||||||||||||||||||||||||||| b. BIX c. F-type d. UPC

9. Which standards were developed to define industry-standard pinouts and color coding for twisted-pair cabling? (Choose two.) a. CWDM b. TIA/EIA-568A c. DWDM d. TIA/EIA-568B 10. What fiber type would you use if the requirements to be met included an SMF media type, 1Gbps bandwidth capacity, and a distance limitation of 5 km? a. 1000BASE-LX b. 1000BASE-SX c. 1000BASE-TX d. Thicknet

Chapter 4: IP Addressing

This chapter covers the following topics related to Objective 1.4 (Given a scenario, configure a subnet and use appropriate IP addressing schemes) of the CompTIA Network+ N10-008 certification exam:

- Public vs. private
    - RFC1918
    - Network address translation (NAT)
    - Port address translation (PAT)
- IPv4 vs. IPv6
    - Automatic Private IP Addressing (APIPA)
    - Extended unique identifier (EUI-64)
    - Multicast
    - Unicast
    - Anycast
    - Broadcast
    - Link local Loopback
    - Default gateway
- IPv4 subnetting
    - Classless (variable-length fixed mask)
    - Classful
        - A
        - B

        - C
        - D
        - E
    - Classless Inter-Domain Routing (CIDR) notation
- IPv6 concepts
    - Tunneling
    - Dual stack
    - Shorthand notation
    - Router advertisement
    - Stateless address autoconfiguration (SLAAC)
- Virtual IP (VIP)
- Subinterfaces

When two devices on a network want to communicate, they need logical addresses (that is, Layer 3 addresses, as described in Chapter 1, “The OSI Model and Encapsulation”). Most modern networks use Internet Protocol (IP) addressing. Therefore, the focus of this chapter is IP. This chapter covers two versions of IP: IP version 4 (IPv4) and IP version 6 (IPv6). First, it discusses how IP concepts apply to IPv4. This discussion introduces you to how IP addresses are represented in binary notation. You will learn about the structure of an IPv4 address and learn to distinguish between different categories of IPv4 addresses. Next, this chapter details various options for assigning IP addresses to end stations. As you will see, one of the benefits of IP addressing is that you have flexibility in how you can subdivide a network address into multiple subnets. This discussion of subnetting is a bit mathematical, and multiple practice exercises are provided to help solidify these concepts in your mind. Although IPv4 is the most widely deployed Layer 3 addressing scheme in today’s networks, its scalability limitation is causing available IPv4 addresses to quickly become depleted. Fortunately, a newer version of IP, |||||||||||||||||||||||||||||||||||||||| IPv6, is scalable beyond anything you will need in your lifetime. So, after focusing on the foundation of IP addressing laid by IPv4, this chapter concludes by introducing you to the fundamental characteristics of IPv6 addressing.

Foundation Topics

Binary Numbering

Chapter 1 describes how a network transmits data as a series of binary 1s and 0s. Similarly, IP addresses are represented as a series of binary digits (that is, *bits*). An IPv4 address consists of 32 bits, and an IPv6 address has a whopping 128 bits. Later in this chapter, you will need to be able to convert between the decimal representation of a number and that number’s binary equivalent. This skill is needed for things such as subnet mask calculations. This section describes this mathematical procedure and provides you with practice exercises.

Principles of Binary Numbering

You are accustomed to using base 10 numbering on a day-to-day basis. In a base 10 numbering system, you have 10 digits, in the range 0 through 9, at your disposal. Binary numbering, however, uses a base 2 numbering system, where there are only two digits: 0 and 1. Because computer systems divide 32-bit IP addresses into four 8-bit octets each, this discussion focuses on converting between 8-bit binary numbers and decimal numbers. To convert a binary number to decimal, you can create a table like Table 4-1.

<key_topic>

Table 4-1 Binary Conversion Table

128	64	32	16	8	4	2	1

Note the structure of this table. There are eight columns, representing the 8 bits in an octet. The column headings are the powers of 2, from 0 to 7, beginning in the rightmost column. Specifically, 2 raised to the power of 0 (20) is 1. (In fact, any number raised to the power of 0 is 1.) If you raise 2 to the first power (21), that equals 2, and 2 raised to the second power (that is, 2 2 , or 2 × 2) is 4. This continues through 2 raised to the power of 7 (that is, 2 7 , or 2 × 2 × 2 × 2 × 2 × 2 × 2), which equals 128. You can use this table for converting binary numbers to decimal and decimal numbers to binary. The skill of binary-to-decimal and decimal-to-binary conversion is critical for working with subnet masks, as discussed later in this chapter.

Converting a Binary Number to a Decimal Number

<key_topic>

To convert a binary number to a decimal number, you populate the previously described binary table with the given binary digits. Then you add up the column heading values for the columns that contain a binary 1. For example, consider Table 4-2. Only the 128, 16, 4, and 2 columns contain a 1, and all the other columns contain a 0. If you add all the column headings containing a 1 in their column (that is, 128 + 16 + 4 + 2), you get the result 150. Therefore, you can conclude that the binary number 10010110 equates to the decimal value 150.

Table 4-2 Binary Conversion Example 1

Table 4-2 Binary Conversion Example 1

128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0

To convert numbers from decimal to binary, staring with the leftmost column, ask the question, “Is this number equal to or greater than the column heading?” If the answer to that question is no, place a 0 in that column and move to the next column. If the answer is yes, place a 1 in that column and subtract the value of the column heading from the number you are converting. When you then move to the next column (to your right), again ask yourself, “Is this number (which is the result of your earlier subtraction) equal to or greater than the column heading?” This process continues (to the right) for all the remaining column headings. For example, say that you want to convert the number 167 to binary. The following steps walk you through the process:

Step 1: Ask the question: “Is 167 equal than or greater to 128”? Because the answer is yes, you place a 1 in the 128 column, as shown in Table 4-3, and subtract 128 from 167, which yields the result 39.

Step 2: Now that you are done with the 128 column, move (to the right) to the 64 column. Ask the question, “Is 39 equal to or greater than 64?” Because the answer is no, you place a 0 in the 64 column, as shown in Table 4-4, and continue to the next column (the 32 column).

Table **4-4:** Binary Conversion Example 2: Step 2

Table 4-4 Binary Conversion Example 2: Step 2

128	64	32	16	8	4	2	1
1	0						

Step 3: Under the 32 column, ask the question, “Is 39 equal to or greater than 32?” Because the answer is yes, you place a 1 in the 32 column, as shown in Table 4-5, and subtract 32 from 39, which yields the result 7.

Table 4-5 Binary Conversion Example 2: Step 3

128	64	32	16	8	4	2	1
1	0	1					

Step 4: Now you are under the 16 column and ask, “Is 7 equal to or greater than 16?” Because the answer is no, you place a 0 in the 16 column, as shown in Table 4-6, and move to the 8 column.

Table 4-6 Binary Conversion Example 2: Step 4

128	64	32	16	8	4	2	1
1	0	1	0				

Step 5: As with the 16 column, the number 7 is not greater than or equal to or greater than 8. So, you place a 0 in the 8 column, as shown in Table 4-7.

Step 6: Because 7 is greater than or equal to 4, you place a 1 in the 4 column, as shown in Table 4-8, and subtract 4 from 7, yielding 3 as the result.

Table 4-8 Binary Conversion Example 2: Step 6

128	64	32	16	8	4	2	1
1	0	1	0	0	1		

Step 7: Now under the 2 column, you ask the question: “Is 3 greater than or equal to 2”? Because the answer is yes, you place a 1 in the 2 column, as shown in Table 4-9, and subtract 2 from 3, yielding 1 as the result.

Table 4-9: Binary Conversion Example 2: Step 7

Table 4-9 Binary Conversion Example 2: Step 7

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	

Step 8. Finally, in the rightmost column (that is, the 1 column), you ask whether the number 1 is greater than or equal to 1. Because it is, you place a 1 in the 1 column, as shown in Table 4-10.

Table 4-10 Binary Conversion Example 2: Step 8

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	1

You can now conclude the decimal number 167 equates to the binary value 10100111. In fact, you can check your work by adding up the values for the column headings that contain a 1 in their column. In this example, the 128, 32, 4, 2, and 1 columns contain a 1. If you add these values, the result is 167 (that is, 128 + 32 + 4 + 2 + 1 = 167).

Binary Numbering Practice

Because binary number conversion is a skill developed through practice, you will now be challenged with a few conversion exercises. The first two exercises ask you to convert a binary number to a decimal number, and the last two exercises ask you to convert a decimal number to a binary number.

Binary Conversion Exercise 1

Using Table 4-11 as a reference, convert the binary number 01101011 to a decimal number.

Table 4-11: Binary Conversion Exercise 1: Base Table

Table 4-11 Binary Conversion Exercise 1: Base Table

128	64	32	16	8	4	2	1

Write your answer here: _____

Binary Conversion Exercise 1: Solution

Given the binary number 01101011 and filling in a binary conversion table, as shown in Table 4-12, you find that the 64, 32, 8, 2, and 1 columns contain a 1. Each of the other columns contains a 0. By adding up the column headings for the columns that contain a 1 (that is, 64 + 32 + 8 + 2 + 1), you get the decimal value 107.

Table 4-12 Binary Conversion Exercise 1: Solution Table

128	64	32	16	8	4	2	1
0	1	1	0	1	0	1	1

Binary Conversion Exercise 2

Using Table 4-13 as a reference, convert the binary number 10010100 to a decimal number.

Table 4-13 Binary Conversion Exercise 2: Base Table

128	64	32	16	8	4	2	1

Write your answer here: _____

Binary Conversion Exercise 2: Solution

Given the binary number 10010100 and filling in a binary conversion table, as shown in Table 4-14, you find that the 128, 16, and 4 columns contain a 1. Each of the other columns contains a 0. By adding up the column headings for the columns that contain a 1 (that is, 128 + 16 + 4), you get the decimal value 148.

Binary Conversion Exercise 3

Using Table 4-15 as a reference, convert the decimal number 49 to a binary number.

Table 4-15 Binary Conversion Exercise 3: Base Table

128	64	32	16	8	4	2	1

Write your answer here: _____

Binary Conversion Exercise 3: Solution

You can begin your conversion of the decimal number 49 to a binary number by asking the following questions and performing the following calculations:

1. Is 49 greater than or equal to 128? No. Put a 0 in the 128 column.
2. Is 49 greater than or equal to 64? No. Put a 0 in the 64 column.
3. Is 49 greater than or equal to 32? Yes. Put a 1 in the 32 column and subtract 32 from 49. 49 – 32 = 17.
4. Is 17 greater than or equal to 16? Yes. Put a 1 in the 16 column and subtract 16 from 17. 17 – 16 = 1.
5. Is 1 greater than or equal to 8? No. Put a 0 in the 8 column.
6. Is 1 greater than or equal to 4? No. Put a 0 in the 4 column.
7. Is 1 greater than or equal to 2? No. Put a 0 in the 2 column.
8. Is 1 greater than or equal to 1? Yes. Put a 1 in the 1 column.

Combining these 8 binary digits, you form the binary number 00110001, as shown in Table 4-16. Verify your work by adding the values of the column headings whose columns contain a 1. In this case, columns 32, 16, and 1 each contain a 1. By adding these values (that is, 32 + 16 + 1), you get the value 49.

value 49.

Table 4-16 Binary Conversion Exercise 3: Solution Table

128	64	32	16	8	4	2	1
0	0	1	1	0	0	0	1

Binary Conversion Exercise 4

Using Table 4-17 as a reference, convert the decimal number 236 to a binary number.

Table 4-17 Binary Conversion Exercise 4: Base Table

128	64	32	16	8	4	2	1

Write your answer here: _____

Binary Conversion Exercise 4: Solution

You can begin your conversion of the decimal number 236 to a binary number by asking the following questions and performing the following calculations:

1. Is 236 greater than or equal to 128? Yes. Put a 1 in the 128 column and subtract 128 from 236. 236 – 128 = 108.

2. Is 108 greater than or equal to 64? Yes. Put a 1 in the 64 column and subtract 64 from 108. 108 – 64 = 44.
3. Is 44 greater than or equal to 32? Yes. Put a 1 in the 32 column and subtract 32 from 44. 44 – 32 = 12.
4. Is 12 greater than or equal to 16? No. Put a 0 in the 16 column.
5. Is 12 greater than or equal to 8? Yes. Put a 1 in the 8 column and subtract 8 from 12. 12 – 8 = 4.
6. Is 4 greater than or equal to 4? Yes. Put a 1 in the 4 column and subtract 4 from 4. 4 – 4 = 0.
7. Is 0 greater than or equal to 2? No. Put a 0 in the 2 column. 8. Is 0 greater than or equal to 1? No. Put a 0 in the 1 column.

By combining these 8 binary digits, you form the binary number 11101100, as shown in Table 4-18. You can verify your work by adding the values of the column headings whose columns contain a 1. In this case, columns 128, 64, 32, 8, and 4 each contain a 1. By adding these values (that is, 128 + 64 + 32 + 8 + 4), you get the value 236.

Table 4-18 Binary Conversion Exercise 4: Solution Table

Table 4-18 Binary Conversion Exercise 4: Solution Table

128	64	32	16	8	4	2	1
1	1	1	0	1	1	0	0

IPv4 Addressing

Although IPv6 is increasingly being adopted in corporate networks, IPv4 is by far the most popular Layer 3 addressing scheme in today's networks. For brevity in this section, the term *IPv4 address* is used interchangeably with the term *IP address*. Devices on an IPv4 network use unique IP addresses to communicate with one another. Metaphorically, you can relate this to sending a letter through the postal service. You place a destination address on an envelope containing the letter, and in the upper-left corner of the envelope, you place your return address. Similarly, when an IPv4 network device sends data on a network, it places both a destination IP address and a source IP address in the packet's IPv4 header.

IPv4 Address Structure

An IPv4 address is a 32-bit address. However, rather than write out each individual bit value, you write the address in *dotted-decimal notation.* Consider the IP address 10.1.2.3. Notice that this IP address is divided into four separate numbers, separated by periods. Each number represents one-fourth of the IP address. Specifically, each number represents an 8-bit portion of the 32 bits in the address. Because each of these four divisions of an IP address represents 8 bits, these divisions are called *octets*. For example, Figure 8-1 shows the binary representation of the 10.1.2.3 IP address. In Figure 4-1, notice that the 8 leftmost bits of 00001010 equate to the decimal value 10. (The calculation for this is described in the previous section.) Similarly, 00000001 in binary equates to 1 in decimal, and 00000010 in binary equals 2 in decimal. Finally, 00000011 yields the decimal value 3.

FIGURE 4-1 Binary Representation of a Dotted-Decimal Address

Interestingly, an IP address is composed of two types of addresses: a network address and a host address. Specifically, a group of contiguous leftjustified bits represent the network address, and the remaining bits (that is, a group of contiguous right-justified bits) represent the address of a host on a network. The IP address component that determines which bits refer to the network and which bits refer to the host is called the *subnet mask*. You can think of the subnet mask as a dividing line separating an IP address's 32 bits into a group of network bits (on the left) and a group of host bits (on the right). A subnet mask typically consists of a series of contiguous 1s followed by a set of contiguous 0s. In total, a subnet mask contains 32 bits, which correspond to the 32 bits found in an IPv4 address. The 1s in a subnet mask correspond to network bits in an IPv4 address, and 0s in a subnet mask correspond to host bits in an IPv4 address. For example, consider Figure 4-2. The 8 leftmost bits of the subnet mask are 1s, and the remaining 24 bits are 0s. As a result, the 8 leftmost bits of the IP address represent the network address, and the remaining 24 bits represent the host address.

Dotted-Decimal Notation	10	1	2	3
IP Address (in Binary)	00001010	00000001	00000010	00000011
Subnet Mask	11111111	00000000	00000000	00000000

Network Bits | Host Bits

FIGURE 4-2 Dividing an IP Address into a Network Portion and a Host Portion

When you write a network address, all host bits are set to 0s. Once again, consider the example shown in Figure 4-2. The subnet mask in this example is an *8-bit subnet mask*, meaning that the 8 leftmost bits in the subnet mask are 1s. If the remaining bits are set to 0, as shown in Figure 4-3, the network address is 10.0.0.0.

When writing a network address, or an IP address for that matter, you need to provide more detail than just a dotted-decimal representation of an IP address's 32 bits. For example, just being told that a device has IP address 10.1.2.3 does not tell you the network on which the IP address resides. To know the network address, you need to know the subnet mask, which could be written in dotted-decimal notation or in ***prefix notation*** (also known as ***slash notation***). In the example with the IP address 10.1.2.3 and an 8-bit subnet mask, the IP address could be written as 10.1.2.3 255.0.0.0 or 10.1.2.3 /8. Similarly, the network address could be written as 10.0.0.0 255.0.0.0 or 10.0.0.0 /8.

Network Address (in Dotted Decimal)	10	0	0	0
Network Address (in Binary)	00001010	00000000	00000000	00000000
Subnet Mask	11111111	00000000	00000000	00000000

Network Bits Host Bits

FIGURE 4-3 Network Address Calculation

Classes of Addresses

Although for an IP address (or a network address) you need subnet mask information to determine which bits represent the network portion of the address, there are default subnet masks with which you should be familiar. The default subnet mask for a given IP address is solely determined by the value in the IP addresses's first octet. Table 4-19 shows the default subnet masks for various ranges of IP addresses.

<key_topic>

Table 4-19 IP Address Classes

Address Class	Value in First Octet	Classful Mask (Dotted Decimal)	Classful Mask (Prefix Notation)
Class A	1–126	255.0.0.0	/8
Class B	128–191	255.255.0.0	/16
Class C	192–223	255.255.255.0	/24
Class D	224–239	—	—
Class E	240–255	—	—

These IP address ranges, which you should memorize, are referred to as different *classes* of addresses. Class A, B, and C addresses are assigned to network devices. Class D addresses

are used as destination IP addresses (that is, not assigned to devices sourcing traffic) for multicast networks, and Class E addresses are reserved for experimental use. The default subnet masks associated with address Classes A, B, and C are called *classful masks*.

For example, consider the IP address 172.16.40.56. If you are told that this address uses its classful mask, you should know that it has subnet mask 255.255.0.0, which is the classful mask for a Class B IP address. You should know that 172.16.50.56 is a Class B IP address, based on the value of the first octet (172), which falls in the Class B range 128-191.

Note

You might have noticed that in the ranges of values in the first octet, the number 127 seems to have been skipped. The reason is that 127 is Technet24 |||||||||||||||||||||||||||||||||||||||| used as a ***loopback*** address, which is a locally significant IP address representing the device itself. For example, if you are working on a network device and want to verify the device has a TCP/IP stack loaded, you can try to ping IP address 127.1.1.1. If you receive ping responses, you can conclude that the device is running a TCP/IP stack.

The nonprofit corporation Internet Corporation for Assigned Names and Numbers (ICANN) globally manages publicly routable IP addresses. ICANN does not directly assign a block of IP addresses to your Internet service provider (ISP) but rather assigns a block of IP addresses to a regional Internet registry. One example of a regional Internet registry is the American Registry for Internet Numbers (ARIN), which acts as an Internet registry for North America. The Internet Assigned Numbers Authority (IANA) is yet another entity responsible for IP address reassignment. The ICANN operates IANA and is responsible for IP address assignment outside North America.

Note

Some literature references the *Internet Network Information Center (InterNIC)*, which was the predecessor to ICANN and existed until September 18, 1998.

When an organization is assigned one or more publicly routable IP addresses by its service provider, that organization often needs more IP addresses to accommodate all of its devices. One solution is to use private IP addressing within an organization, in combination with ***Network Address Translation (NAT)***. Remember, specific Class A, B, and C networks have been designated for private use. Although these networks are routable (with the exception of the 169.254.0.0–169.254.255.255 address range) within the organization, ISPs do not route these private networks over the public Internet. Table 4-20 shows the IP networks reserved for internal use.

<key_topic>

Table 4-20 Private IP Networks

Address Class	Address Range	Default Subnet Mask
Class A	10.0.0.0–10.255.255.255	255.0.0.0
Class B	172.16.0.0–172.31.255.255	255.255.0.0
Class B	169.254.0.0–169.254.255.255	255.255.0.0
Class C	192.168.0.0–192.168.255.255	255.255.255.0

Note

The 169.254.0.0–169.254.255.255 address range is not routable. Addresses in this range are only usable on their local subnet and are dynamically assigned to network hosts using Automatic Private IP Addressing (APIPA), which is discussed later in this chapter.

NAT, which is available on routers, allows private IP addresses used within an organization to be translated into a pool of one or more publicly routable IP addresses.

Types of Addresses

For the real world and for the Network+ exam, you need to be familiar with the following categories of IPv4 addresses (and even more, which we will discuss later): unicast, broadcast, and multicast. The following sections describe these types of addresses in detail.

Unicast

Most network traffic is *unicast* in nature, meaning that traffic travels from a single source device to a single destination device. Figure 4-4 illustrates an example of a unicast transmission.

example of a unicast transmission.

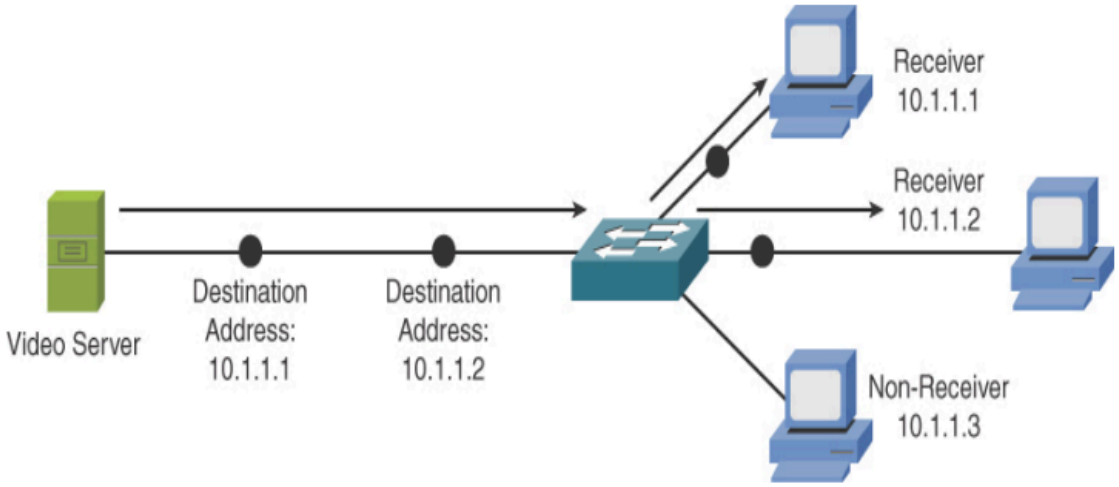


FIGURE 4-4 Sample Unicast Transmission

Broadcast

Broadcast traffic travels from a single source to all destinations on a network (that is, a *broadcast domain*). It might seem as though the broadcast address 255.255.255.255 would reach all hosts on all interconnected networks. However, 255.255.255.255 targets all devices on a single network—specifically, the network local to the device sending a packet destined for 255.255.255.255. Another type of broadcast address is a *directed broadcast address,* which targets all devices in a remote network. For example, the address 172.16.255.255 /16 is a directed broadcast address targeting all devices in the 172.16.0.0 /16 network. Figure 4-5 illustrates an example of a broadcast transmission.

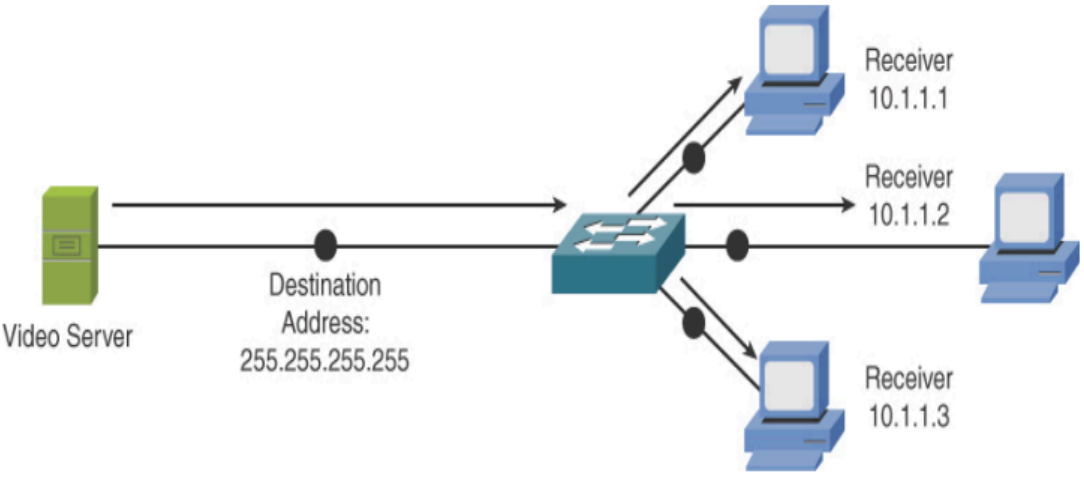


FIGURE 4-5 Sample Broadcast Transmission

Multicast

Multicast technology offers an efficient mechanism for a single host to send traffic to multiple specific destinations. For example, say that a network has 100 users, and 20 of those users want to receive a video stream from a video server. With a unicast solution, the video server would have to send 20 individual streams, 1 for each recipient. Such a solution could consume a significant amount of network bandwidth and put a heavy processor burden on the video server.

With a broadcast solution, the video server would only have to send the video stream once; however, it would be received by every device on the local subnet, even devices not wanting to receive the video stream. Even though a lot of the devices do not want to receive the video stream, they still must pause what they are doing and take time to check each of these unwanted packets. As shown in Figure 4-6, multicast offers a compromise, allowing the video server to send the video stream only once and sending the video stream only to devices on the network that want to receive the stream. Multicast is possible thanks to the use of a Class D address. A Class D address, such as 239.1.2.3, represents the address of a *multicast group*. The video server could, in this example, send a single copy of each video packet destined for 239.1.2.3. Devices wanting to receive the video stream could Technet24 ||||||||||||||||||||||||||||||||||||||||||||||||| join the multicast group. Based on the device request, switches and routers in the topology could then dynamically determine out of which ports the video stream should be forwarded.

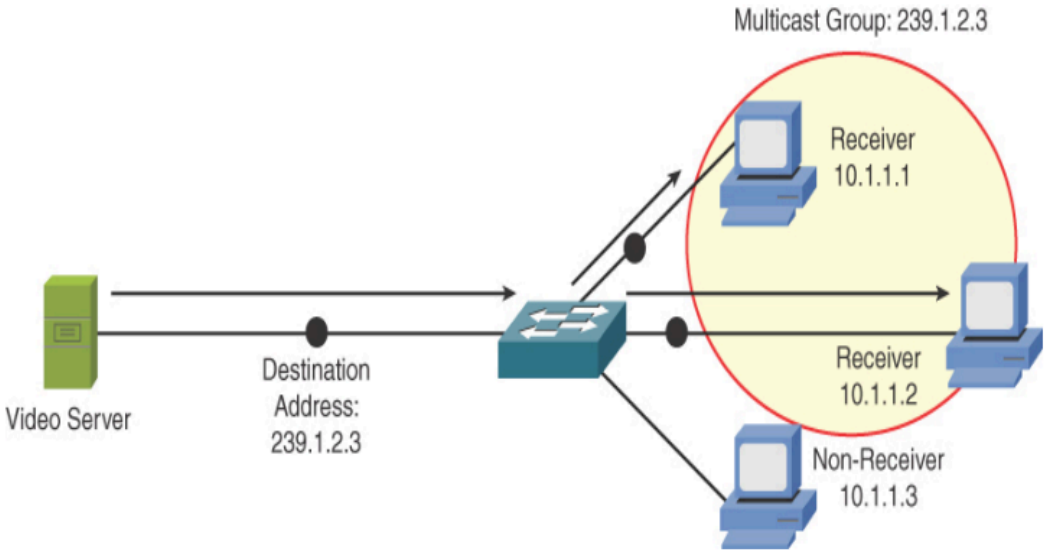


FIGURE 4-6 Sample Multicast Transmission

Assigning IPv4 Addresses

At this point in the discussion, you should understand that networked devices need IP addresses. However, beyond just an IP address, what extra IP address–related information needs to be provided to a device, and how does an IP address get assigned to a device?

IP Addressing Components

As discussed in the previous section, an IP address has two portions: a network portion and a host portion. A subnet mask is required to delineate between these two portions. In addition, if traffic is destined for a different subnet on which the traffic originates, a ***default gateway*** needs to be defined. A |||||||||||||||||||||||||||||||||||||||||| default gateway routes traffic from the sender's subnet toward the destination subnet. Chapter 10, "Routing Technologies and Bandwidth Management," covers the concept of routing.

Another consideration is that end users do not typically type in the IP address of the destination device with which they wish to connect (for example, a web server on the Internet). Instead, end users typically type in fully qualified domain names (FQDNs), such as www.ajsnetworking.com. When connecting to devices on the public Internet, a Domain Name System (DNS) server translates an FQDN into the corresponding IP address. For a very long time, in a company's internal network (that is, an intranet), a Microsoft Windows Internet Name Service (WINS) was used to convert the names of network devices into their corresponding IP addresses. For example, say that you attempted to navigate to the shared folder \\server1\hrdocs. A WINS server could be used to resolve the network device name server1 to a corresponding IP address. The path \\server1\hrdocs is in *universal naming convention (UNC)* form, where you are specifying a network device name (in this case, server1) and a resource available on that device (in this case, hrdocs). Companies today use DNS even for internal network name resolution. To summarize, network devices (for example, an end-user PC) can benefit from a variety of IP address parameters, such as the following:

- IP address
- Subnet mask
- Default gateway
- Server address

Remember as well that an IP address no longer needs to be assigned to a single entity or interface. A *virtual IP (VIP) address* is commonly used and can fulfill many purposes, such as the following:

- Provide key addresses used in address translation.
- Represent any actual IP address assigned to a network device interface.
- Permit the sending of traffic to multiple different network devices, all configured to respond based on the virtual IP address.

A concept that is very similar to the public IP address is the ***subinterface***. This is a handy interface capability supported by most router and switch manufacturers that allows you to create

many virtual interfaces out of a single physical interface. Example 4-1 demonstrates the creation of a subinterface on a Cisco router and the assignment of an IP address there. This is done after enabling and providing an IP address to the physical interface.

Example 4-1 Creating a Subinterface on a Router and Assigning an IP Address

Click here to view code image

```
interface gi0/0
no shutdown
ip address 10.10.10.10 255.255.255.0
!
interface gi0/0.100
ip address 10.100.100.10 255.255.255.0
```

Note

If a physical interface is enabled and in an up/up state, the subinterface should also be enabled, and its IP address should be functional on the network.

Static Configuration

A simple way of configuring a PC with, for example, IP address parameters is to statically configure that information. For example, on a PC running Microsoft Windows as the operating system, you can navigate to the Control Panel, as shown in Figure 4-7, and click **Network and Internet.**

In the Network and internet control panel, click ***Network and Sharing Center***, as shown in Figure 4-8.

You can then click the ***Change adapter settings*** link, as shown in Figure 4-9

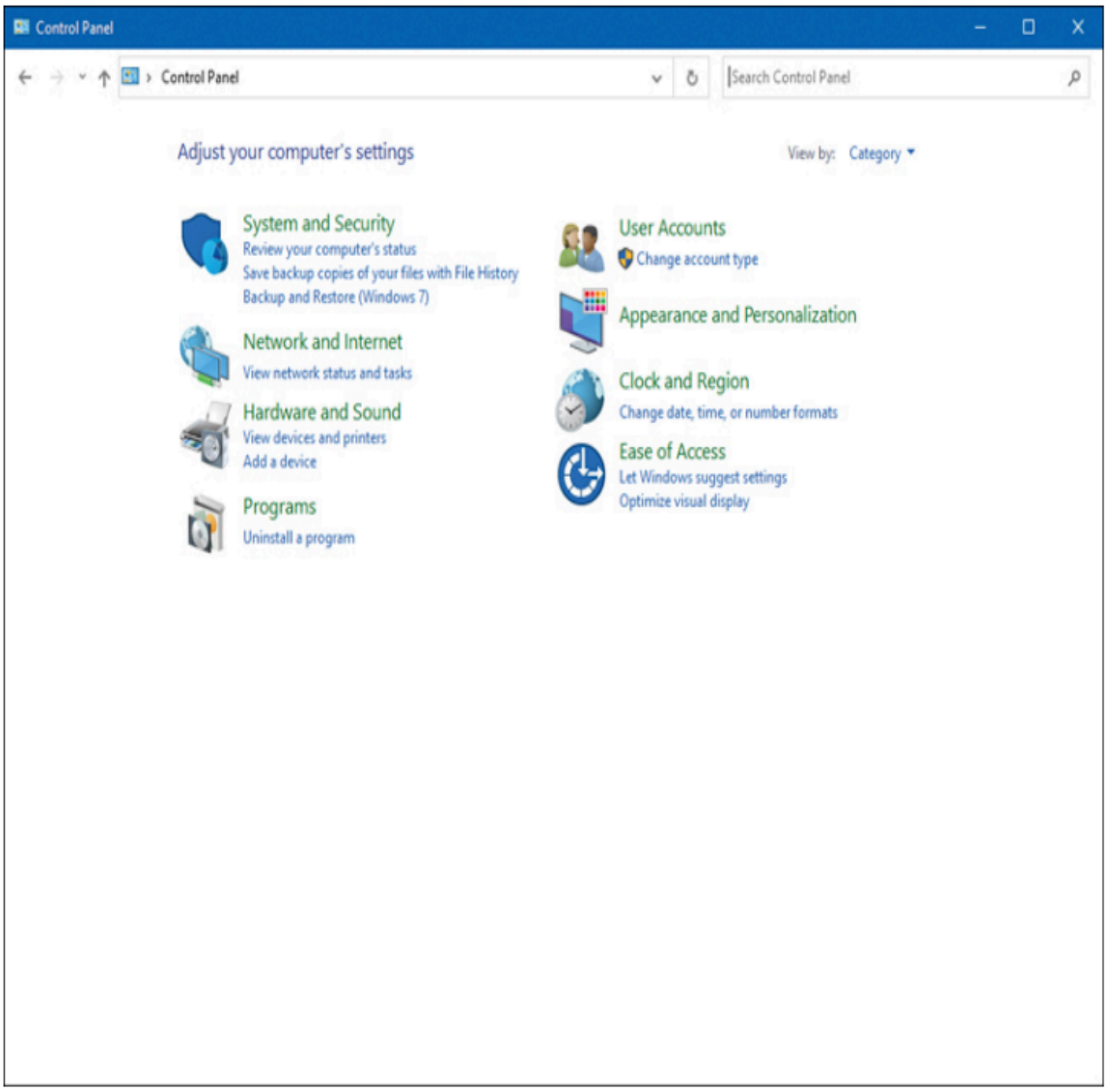


FIGURE 4-7 Windows Control Panel

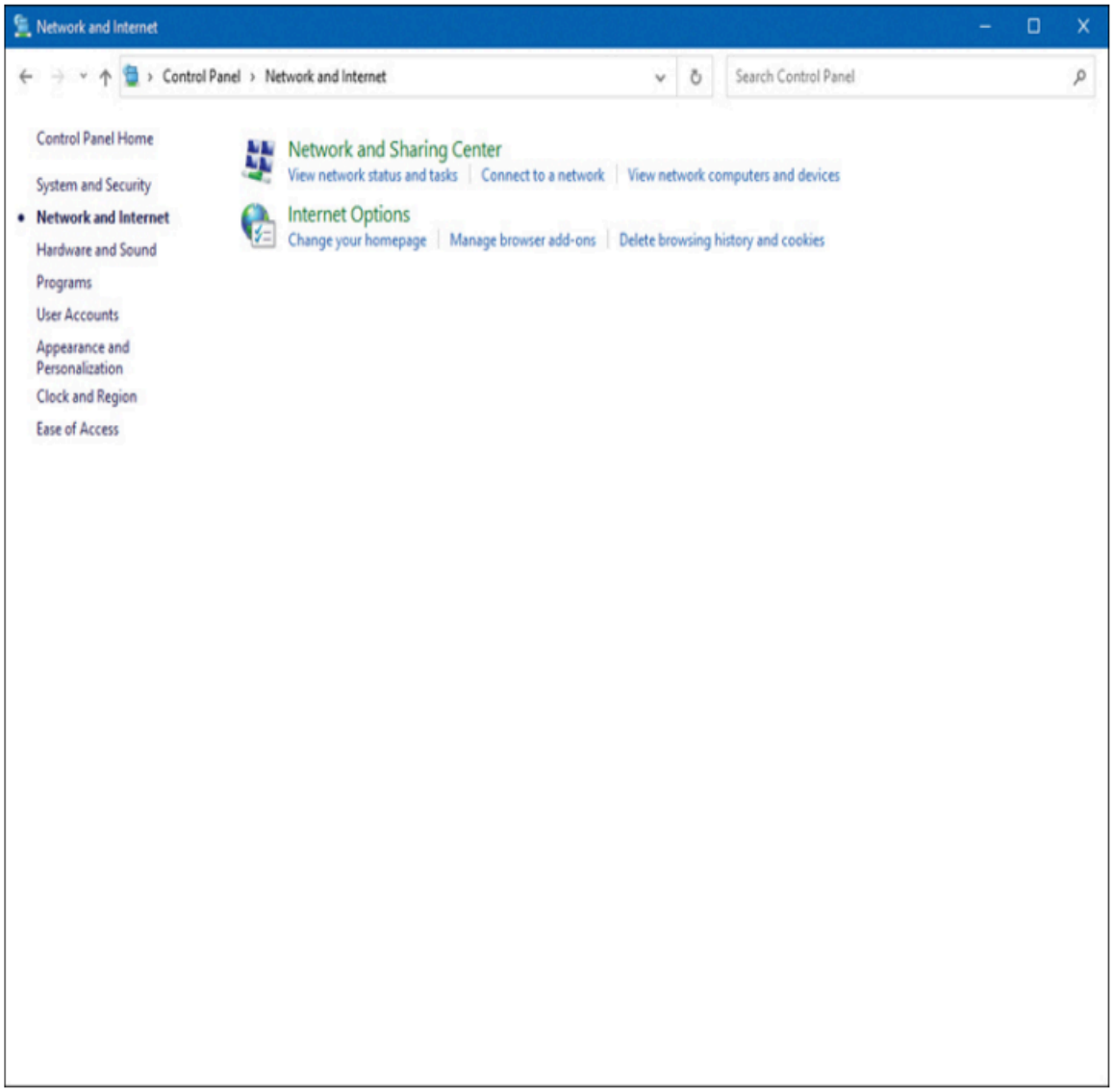


FIGURE 4-8 Network and Internet Control Panel

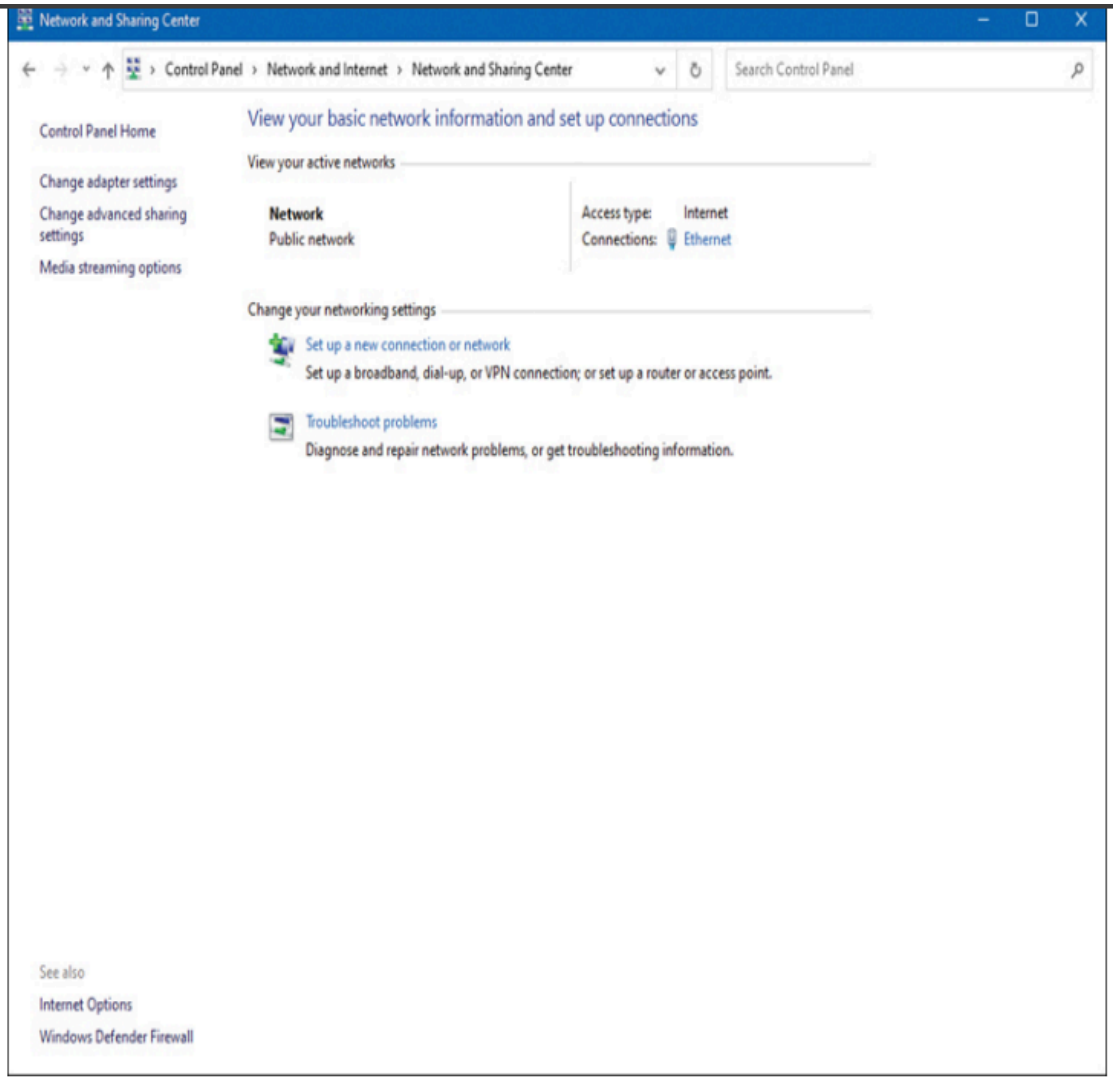


FIGURE 4-9 Network and Sharing Center

In the ***Network Connections*** window, double-click the network adapter whose settings you want to change, as shown in Figure 4-10.

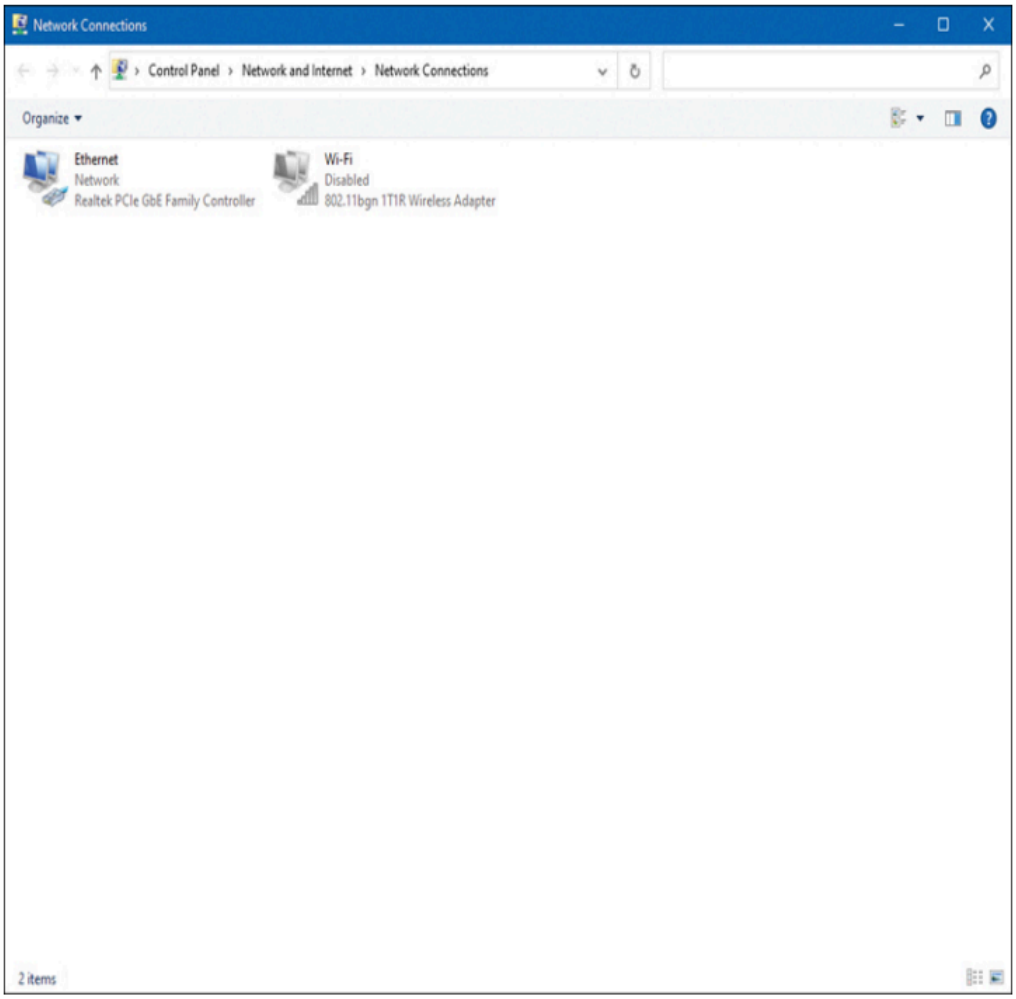


FIGURE 4-10 Network Connections Window

You are then taken to the Local Area Connection Status window, as shown in Figure 4-11, where you can click the ***Properties*** button.

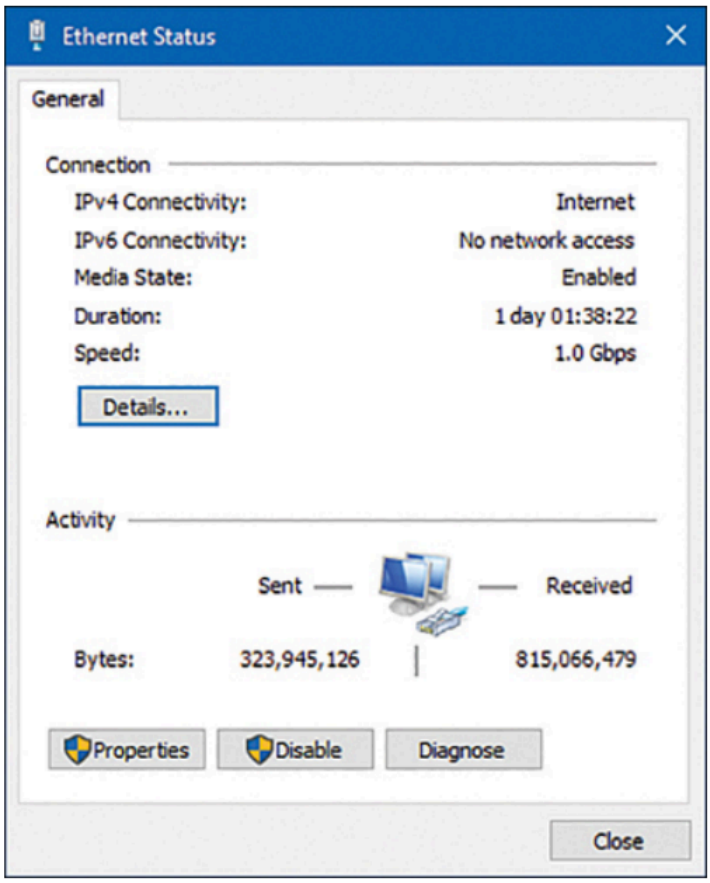


FIGURE 4-11 Local Area Connection Status Window

As shown in Figure 4-12, you can highlight ***Internet Protocol Version 4 (TCP/IPv4)*** and click the ***Properties*** button.

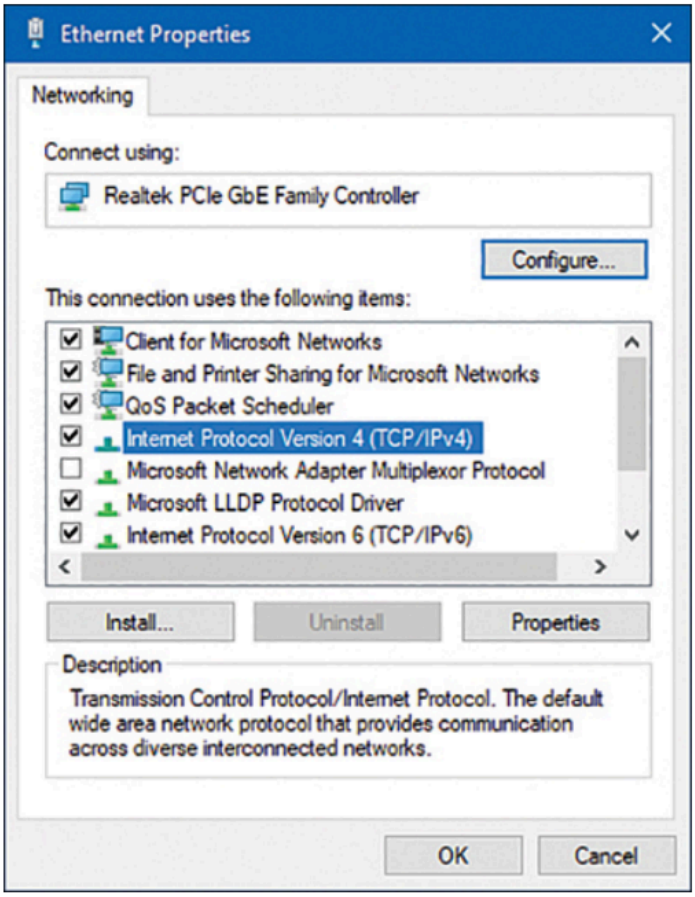


FIGURE 4-12 Local Area Connection Properties

You can enter an IP address, a subnet mask, a default gateway, and DNS server information into the Internet Protocol Version 4 (TCP/IPv4) Properties window, shown in Figure 4-13. Although DNS server information is entered in this window, more advanced DNS options and WINS options are available by clicking the **Advanced** button.

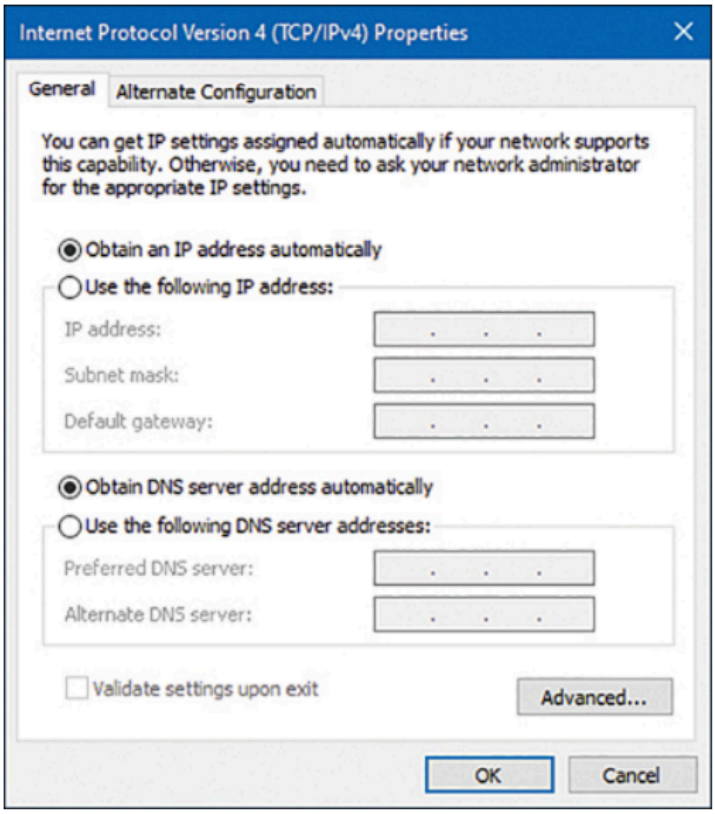


FIGURE 4-13 Internet Protocol Version 4 (TCP/IPv4) Properties

By clicking the ***DNS*** tab in the Advanced TCP/IP settings window, as shown in Figure 4-14, you can add, remove, or reorder DNS servers, and you can adjust various other DNS parameters. Recall that a DNS server converts an FQDN to an IP address. Also, although Figure 4-13 shows the same IP address for the default gateway and a DNS server, these are not always located on the same device. Similarly, you can configure Windows Internet Name Service (WINS) servers in the WINS tab of the Advanced TCP/IP Settings window, as shown in Figure 4-15. Much like a DNS server, a WINS server converts a NetBIOS computer name to a corresponding IP address.

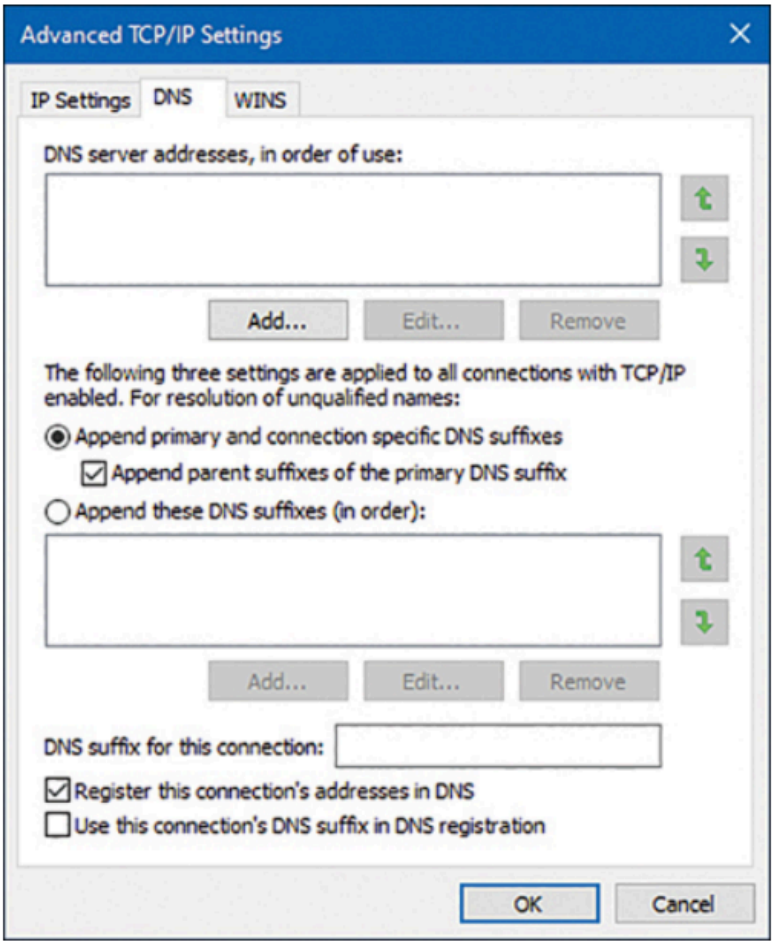


FIGURE 4-14 Advanced TCP/IP Settings: DNS Tab

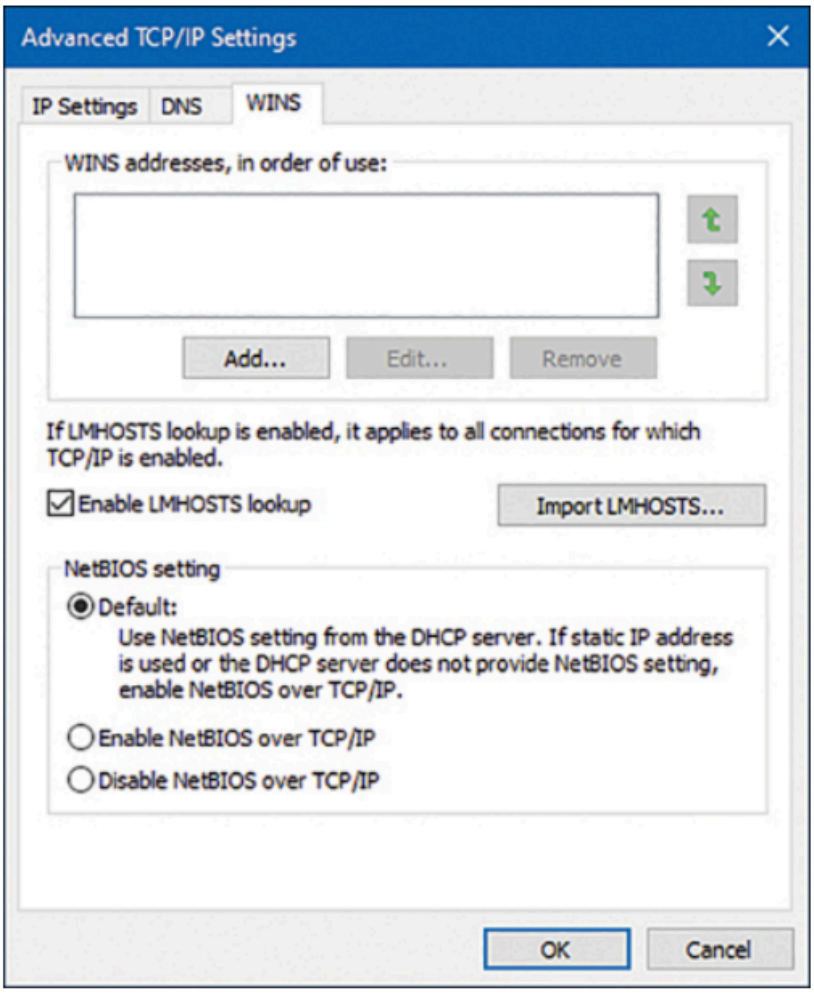


FIGURE 4-15 Advanced TCP/IP Settings: WINS Tab

Dynamic Configuration

Statically assigning IP address information related to individual networked devices can be time-consuming, error-prone, and lacking in scalability. Instead of using static IP address assignments, many corporate networks dynamically assign IP address parameters to their devices. An early choice for performing this automatic assignment of IP addresses was the ***Bootstrap Protocol (BOOTP)***. Currently, however, the most popular approach for dynamic IP address assignment is ***Dynamic Host Configuration Protocol (DHCP)***.

BOOTP

Engineers developed BOOTP as a method of assigning IP address, subnet mask, and default gateway information to diskless workstations. In the early days of Microsoft Windows (for example, Microsoft Windows 3.1), Microsoft Windows did not natively support TCP/IP. To include TCP/IP support, an add-on TCP/IP application (for example, Trumpet Winsock) could be

run. Such an application would typically support BOOTP. When a device needed to obtain IP address information, a BOOTP broadcast would be sent out from the device needing an IP address. If a BOOTP server (BOOTPS) received the broadcast, it could match the source MAC address in the received frame (the MAC address from the device wanting to obtain an IP address) with a corresponding IP address in a database stored on the BOOTP server. The BOOTPS would then respond to the requesting client with IP address information. Because BOOTP requests were based on broadcasts, by default, a BOOTP request could not propagate beyond a device's local subnet. However, most enterprise-class routers can be configured to forward selected broadcast types, including BOOTP broadcasts.

DHCP

DHCP offers a more robust solution to IP address assignment than does BOOTP. DHCP does not require a statically configured database of MAC address-to-IP-address mappings. Also, DHCP has a wide variety of options beyond basic IP address, subnet mask, and default gateway parameters. For example, a DHCP server can educate a DHCP client about the IP address of a TFTP server from which a configuration file could be downloaded. In Chapter 6, "Network Services," you will learn more about the operation of DHCP. For now, it is important to realize that, as with BOOTP, DHCP's initial request is a broadcast, requiring a client's local router be configured to appropriately forward DHCP requests to a DHCP server if that DHCP server is not on the local subnet of the requesting client.

In setting up a DHCP server, you would identify a range of IP addresses to hand out, and this would be referred to as the *scope*. In addition, a DHCP server can be configured to have reservations, so that a specific IP address is reserved for a specific Layer 2 Ethernet MAC address. The lease time can also be configured and is usually set to one day. The DHCP server also provides options such as DNS server address, the default gateway to use, domain suffixes to use, and more. If a DHCP client is not on the same subnet as a DHCP server, a router or another device that is connected to the same subnet as the DHCP client can be configured as a DHCP relay agent. The device takes the discover packet from the client (broadcast) and routes it to the DHCP server (unicast). This feature is also sometimes referred to as *IP helper*.

As an example of DHCP client configuration, in Microsoft Windows 10, you can select the ***Obtain an IP address automatically*** and ***Obtain DNS server address automatically*** options in the Internet Protocol Version 4 (TCP/IPv4) Properties window, as shown in Figure 4-16.

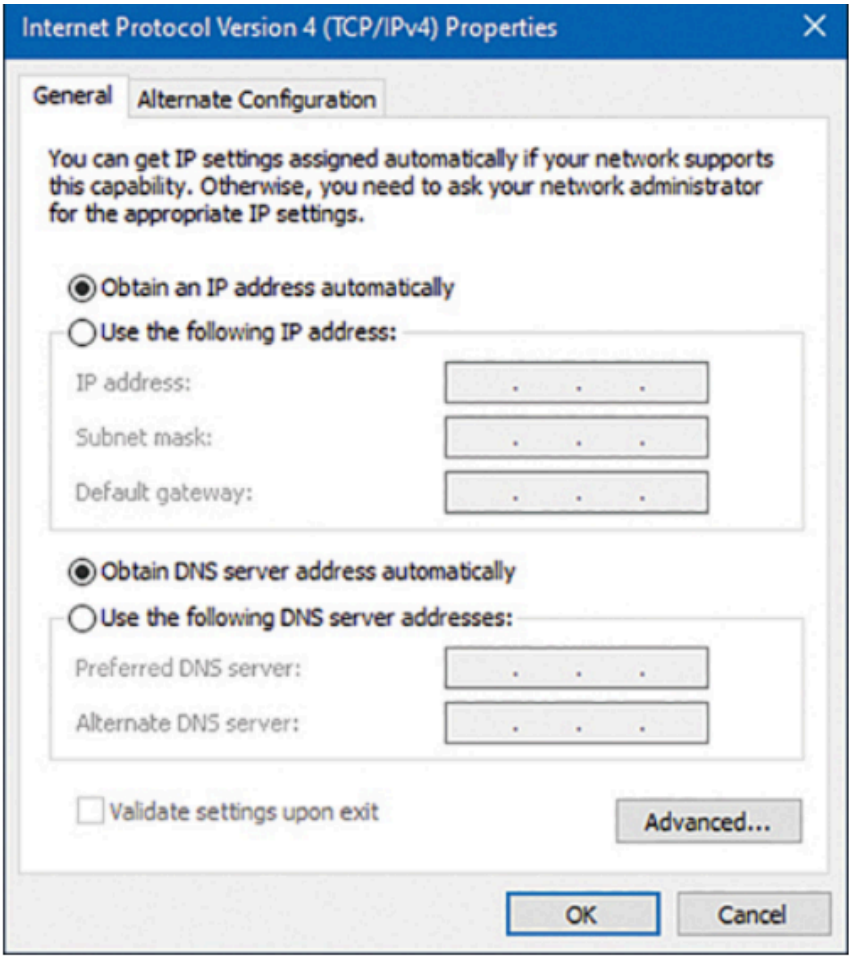


FIGURE 4-16 Configuring Microsoft Windows 10 to Obtain IP Address Information via DHCP

Note

A protocol made obsolete by BOOTP and DHCP is Reverse Address Resolution Protocol (RARP). Whereas Address Resolution Protocol (ARP) requests a MAC address that corresponds to a known IP address, RARP requested an IP address (from a preconfigured host) that corresponded to a station's MAC address. Although RARP did allow a station to dynamically obtain an IP address, both BOOTP and DHCP offer additional features.

Automatic Private IP Addressing

If a networked device does not have a statically configured IP address and is unable to contact a DHCP server, it still might be able to communicate on an IP network thanks to ***Automatic Private IP Addressing (APIPA)***. The APIPA feature allows a networked device to self-assign

an IP address from the 169.254.0.0/16 network. Note that this address is usable only on the device's local subnet. (The IP address is not routable.)

As shown in Figure 4-17, Microsoft Windows 10 defaults to APIPA if a client is configured to automatically obtain IP address information and that client fails to obtain IP address information from a DHCP server.

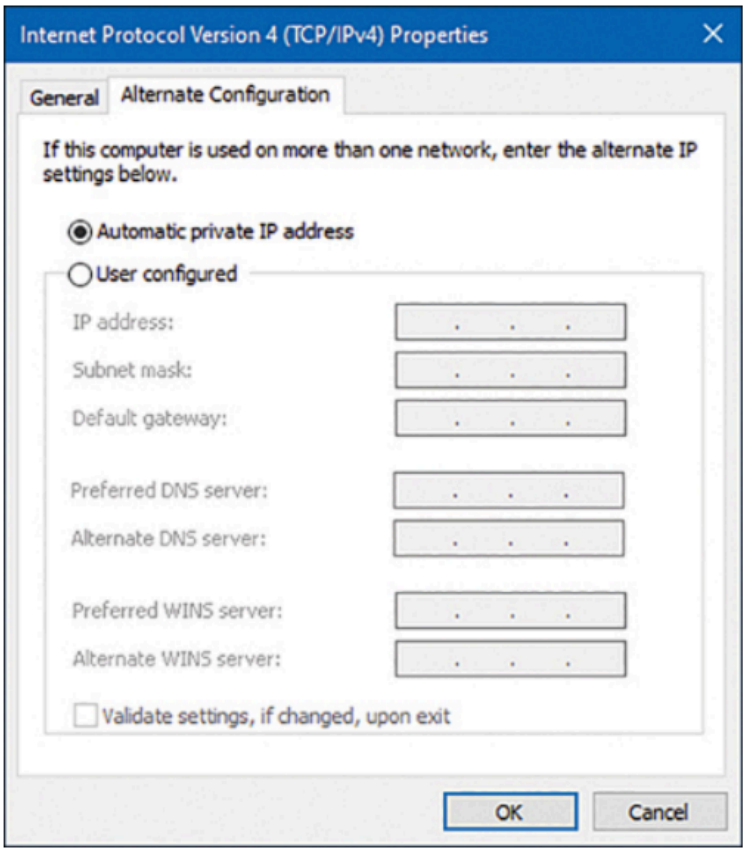


FIGURE 4-17 APIPA Configuration Enabled by Default

APIPA was designed as a solution for quickly setting up a localized network without the need to configure a DHCP server or the need to statically assign IP address information. However, there remains a need for devices on this localized network to perform name resolution and discover network services. Fortunately, these needs are covered by Zero Configuration (Zeroconf). ***Zeroconf*** is a technology supported on most operating systems and performs three basic functions:

<key_topic>

- **Assigning link-local IP addresses:** A ***link-local*** IP address is a nonroutable IP address usable only on a local subnet. APIPA is an example of a technology that assigns link-local IP addresses.
- **Resolving computer names to IP addresses:** Multicast Domain Name System (mDNS) is an example of a technology that can resolve computer names to their corresponding IP addresses on a local subnet, without the aid of a DNS server or a WINS server.
- **Locating network services:** Examples of service discovery protocols include the standards-based Service Location Protocol (SLP), Microsoft's Simple Service Discovery Protocol (SSDP), and Apple's DNS-based Service Discovery (DNS-SD).

If devices supporting these three Zeroconf features are interconnected on a local subnet, they can dynamically obtain link-local IP addresses, resolve one another's names to IP addresses, and discover services available on a network.

Subnetting

Earlier in this chapter, you were introduced to the purpose of a subnet mask and the default subnet masks for the various IP address classes. Default subnet masks (that is, classful subnet masks) are not always the most efficient choice. Fortunately, you can add additional network bits to a subnet mask (thereby extending the subnet mask) to create subnets within a classful network. This section explains why you might want to perform this process and describes how you mathematically perform subnet calculations.

Purpose of Subnetting

Consider the number of assignable IP addresses in the various classes of IP addresses shown in Table 4-21. Recall that the host bits of an IP address cannot be all 0s (which represents the network address) or all 1s (which represents the directed broadcast address). Therefore, the number of assignable IP addresses in a subnet can be determined by the following formula:

Number of assignable IP addresses in a subnet: $2^{h}-2$

where h is the number of host bits in a subnet mask.

<key_topic>

Table 4-21 Assignable IP Addresses

Address Class	Assignable IP Addresses
Class A	16,777,214 (2^{24}–2)
Class B	65,534 (2^{16}–2)
Class C	254 (2^{8}–2)

Suppose that you decide to use a private Class B IP address (for example, 172.16.0.0/16) for your internal IP addressing. For performance reasons, you would not want to support as many as 65,534 hosts in a single broadcast domain. Therefore, a best practice with such a network address is to subnet the network (thereby extending the number of network bits in the network's subnet mask) into additional subnetworks. In fact, you could subnet your major network address space and then further subnet one of your unused subnet addresses!

This practice, known as ***Variable-Length Subnet Masking (VLSM)***, allows you to design the network the best way possible in terms of the number of IP addresses required in different areas. Of course, this network also uses a variety of subnet masks to accomplish this task.

Subnet Mask Notation

As previously mentioned, the number of bits in a subnet mask can be represented in dotted-decimal notation (for example, 255.255.255.0) or in prefix notation (for example, /24). For example, Table 4-22 shows valid subnet masks in dotted-decimal notation and the corresponding prefix notation.

<key_topic>

Table 4-22 Dotted-Decimal and Prefix-Notation Representations for IPv4 Subnets

Dotted-Decimal Notation	Prefix Notation
255.0.0.0	/8 (classful subnet mask for Class A networks)
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16 (classful subnet mask for Class B networks)
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22

255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24 (classful subnet mask for Class C networks)
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28

Dotted-Decimal Notation	**Prefix Notation**
255.255.255.248	/29
255.255.255.252	/30

Recall that any octet with a value of 255 contains eight 1s. Also, you should memorize valid octet values for an octet and the corresponding number of 1s (that is, continuous, left-justified 1s) in that octet, as shown in Table 4- 23. Based on this information, you should be able to see the dotted-decimal notation of a subnet mask and quickly determine the corresponding prefix notation.

<key_topic>

Table 4-23 Subnet Octet Values

Subnet Octet Value	Number of Contiguous Left-Justified Ones
0	0
128	1
192	2
224	3
240	4
248	5
252	6
254	7
255	8

For example, consider the subnet mask 255.255.192.0. Because each of the octets have a value of 255, you know that you have sixteen 1s from the first two octets. You then recall that a value of 192 in the third octet requires two 1s from that octet. By adding sixteen 1s from the first two octets to the two 1s from the third octet, you can determine that the subnet mask 255.255.192.0 has the corresponding prefix notation /18.

To help develop the skill of making these calculations quickly, work through the following two exercises.

Subnet Notation: Practice Exercise 1

Given the subnet mask 255.255.255.248, what is the corresponding prefix notation? _____

Subnet Notation: Practice Exercise 1 Solution

Given the subnet mask 255.255.255.248, you should recognize that the first three octets, each containing a value of 255, represent twenty-four 1s. To those twenty-four 1s, you add five additional 1s, based on your memorization of how many contiguous, left-justified 1s in an octet are required to produce various octet values. The sum of 24 bits (from the first three octets) and the 5 bits (from the fourth octet) gives you a total of 29 bits. Therefore, you can conclude that a subnet mask with dotted-decimal notation 255.255.255.248 has equivalent prefix notation /29.

Subnet Notation: Practice Exercise 2

Given the subnet mask /17, what is the corresponding dotted-decimal notation? _____

Subnet Notation: Practice Exercise 2 Solution

You know that each octet contains 8 bits. So, given the subnet mask /17, you can count by 8s to determine that there are eight 1s in the first octet, eight 1s in the second octet, and one 1 in the third octet. You already knew that an octet containing all 1s has the decimal value 255. From that knowledge, you conclude that each of the first two octets has the value 255. Also, based on your memorization of Table 4-23, you know that one 1 (that is, a left-justified 1) in an octet has the decimal equivalent value 128.

Therefore, you can conclude that a subnet mask with prefix notation /17 can be represented in dotted-decimal notation as 255.255.128.0.

Extending a Classful Mask

The way to take a classful mask (that is, a network using a classful mask) and divide that network into multiple subnets is by adding 1s to the network's classful subnet mask. However, the class of the IP address does not change, regardless of the new subnet mask. For example, if you took the 172.16.0.0/16 network and subnetted it into multiple networks using a 24-bit subnet mask (172.16.0.0/24, 172.16.1.0/24, 172.16.2.0/24, …), those networks would still be Class B networks.

Specifically, the class of a network is entirely determined by the value of the first octet. The class of a network has nothing to do with the number of bits in a subnet, which makes this an often-misunderstood concept. For example, the network 10.2.3.0/24 has the subnet mask of a Class C network (that is, a 24-bit subnet mask). However, the 10.2.3.0/24 network is a Class A network because the value of the first octet is 10. It is simply a Class A network that happens to have a 24-bit subnet mask.

Borrowed Bits

When you add bits to a classful mask, the bits you add are referred to as ***borrowed bits***. The number of borrowed bits you use determines how many subnets are created and the number of usable hosts per subnet.

Calculating the Number of Created Subnets

To determine the number of subnets created when adding bits to a classful mask, you can use the following formula:

<key_topic>

Number of created subnets = 2^s

Where s is the number of borrowed bits.

For example, let's say you subnetted the 192.168.1.0 network with a 28-bit subnet mask, and you want to determine how many subnets were created. First, you determine how many borrowed bits you have. Recall that the number of borrowed bits is the number of bits you have in a subnet mask beyond the classful mask. In this case, because the first octet in the network address has the value 192, you can conclude that this is a Class C network. Also recall that a Class C network has 24 bits in its classful (that is, its default) subnet mask. Because you now have a 28-bit subnet mask, the number of borrowed bits can be calculated as follows:

<key_topic>

Number of borrowed bits = Bits in custom subnet mask – Bits in classful subnet mask Number of borrowed bits = 28 – 24 = 4 Now that you know you have 4 borrowed bits,

You can raise two to the power of four (2^4 or 2 x 2 x 2 x 2), which equals 16. From this calculation, you conclude that subnetting 192.168.1.0/24 with a 28-bit subnet mask yields 16 subnets.

Calculating the Number of Available Hosts

Earlier in this section, you saw the formula for calculating the number of available (that is, assignable) host IP addresses, based on the number of host bits in a subnet mask. Here again is the formula:

<key_topic>

Number of assignable IP address in a subnet = 2*h*-2

where h is the number of host bits in the subnet mask. Using the previous example, let's say you want to determine the number of available host IP addresses in one of the 192.168.1.0/28 subnets. First, you need to determine the number of host bits in the subnet mask. Because you that the an IPv4 address consists of 32 bits, you can subtract the number of bits in the subnet mask (28, in this example) from 32 to determine the number of host bits:

Number of host bits = 32 - Number of bits in subnet mask
Number of host bits = 32-28 = 4

Now that you know the number of host bits, you can apply it to the previously presented formula:

Number of assignable IP addresses in a subnet = 2h-2, where ***h*** is the number of host bits in the subnet mask.

Number of assignable IP addresses in a subnet = 2^4 - 2 = 14

From this calculation, you can conclude that each of 192.168.1.0/28 subnets has 14 usable IP addresses. To reinforce your skill with these calculations, you are now challenged with a few practice exercises.

Basic Subnetting Practice: Exercise 1

Using a separate sheet of paper, solve the following scenario:

> Your company has been assigned the 172.20.0.0/16 network for use at one of its sites. You need to use a subnet mask that will accommodate 47 subnets while simultaneously accommodating the maximum number of hosts per subnet. What subnet mask will you use?

Basic Subnetting Practice: Exercise 1 Solution

To determine how many borrowed bits are required to accommodate 47 subnets, you can write out a table that lists the powers of 2, as shown in Table 4-24. In fact, you might want to sketch out a similar table on the dry-erase card you are given when you take the Network+ exam.

<key_topic>

Table 4-24 Number of Subnets Created by a Specified Number of Borrowed Bits

Borrowed Bits	Number of Subnets Created (2^s, Where *s* Is the Number of Borrowed Bits)
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

In this example, where you want to support 47 subnets, 5 borrowed bits are not enough, and 6 borrowed bits are more than enough. Because 5 borrowed bits are not enough, you round up and use 6 borrowed bits. The first octet in the network address 172.20.0.0 has the value 172, which means you are dealing with a Class B address. Because a Class B address has 16 bits in its classful mask, you can add the 6 borrowed bits to the 16- bit classful mask, which results in a 22-bit subnet mask. One might argue that although a 22-bit subnet mask would accommodate 47 subnets, so would a 23-bit subnet mask or a 24-bit subnet mask. Although that is true, recall that the scenario said you should have the maximum number of hosts per subnet. This suggests that you should not use more borrowed bits than necessary. Therefore, you can conclude that to meet the scenario's requirements, you should use a subnet mask of /22, which could also be written as 255.255.252.0

Basic Subnetting Practice: Exercise 2

Using a separate sheet of paper, solve the following scenario:

Your company has been assigned the 172.20.0.0/16 network for use at one of its sites. You need to calculate a subnet mask that will accommodate 100 hosts per subnet while maximizing the number of available subnets. What subnet mask will you use?

Basic Subnetting Practice: Exercise 2 Solution

To determine how many host bits are required to accommodate 100 hosts, you can write out a table that shows the number of hosts supported by a specific number of hosts bits, as shown in Table 4-25. As with the previous example, you might want to sketch out a similar table on the dry-erase card you are given when taking the Network+ exam.

<key_topic>

Table 4-25 Number of Supported Hosts, Given a Specified Number of Host Bits

Host Bits	Number of Supported Hosts ($2^h - 2$, Where *h* Is the Number of Host Bits)
2	2
3	6
4	14
5	30
6	62
7	126

Host Bits	Number of Supported Hosts ($2^h - 2$, Where *h* Is the Number of Host Bits)

8	254
9	510
10	1022
11	2046
12	4094

In this example, where you want to support 100 hosts, 6 host bits are not enough, and 7 host bits are more than enough. Because 6 host bits are not enough, you round up and use 7 host bits.

Because an IPv4 address has 32 bits and you need 7 host bits, you can calculate the number of subnet bits by subtracting the 7 host bits from 32 (that is, the total number of bits in an IPv4 address). This results in a 25-bit subnet mask (that is, 32 total bits – 7 host bits = 25 subnet mask bits). Therefore, you can conclude that to meet the scenario's requirements, you should use a subnet mask of /25, which could also be written as 255.255.255.128.

Calculating New IP Address Ranges

Now that you can calculate the number of subnets created based on a given number of borrowed bits, the next logical step is to calculate the IP address ranges making up those subnets. For example, if you subnetted 172.25.0.0/16 with a 24-bit subnet mask, the resulting subnets would be as follows:

172.25.0.0/24

172.25.1.0/24

172.25.2.0/24

...

172.25.255.0/24

Let's consider how such a calculation is performed. Notice in the previous example that you count by 1 in the third octet to calculate the new networks. To decide in what octet you start counting and by what increment you count, a new term needs to be defined. The *interesting octet* is the octet that contains the last 1 in the subnet mask. In this example, the subnet mask is

a 24-bit subnet mask, which has the dotted-decimal equivalent 255.255.255.0 and the binary equivalent 11111111.11111111.11111111.00000000. From any of these subnet mask representations, you can determine that the third octet is the octet to contain the last 1 in the subnet mask. Therefore, you will be changing the value of the third octet to calculate the new networks.

Now that you know that the third octet is the interesting octet, you need to know by what increment you will be counting in that octet. This increment is known as the ***block size***, and you can calculate it by subtracting the subnet mask value in the interesting octet from 256. In this example, the subnet mask has the value 255 in the interesting octet (that is, the third octet). If you subtract 255 from 256, you get the result 1 (that is, 256 – 255 = 1). The first subnet is the original network address, with all of the borrowed bits set to 0. After this first subnet, you start counting by the block size (1 in this example) in the interesting octet to calculate the remainder of the subnets.

The process just described for calculating subnets can be described as follows:

<key_topic>

Step 1. Determine the interesting octet by determining the last octet in the subnet mask to contain a 1.

Step 2. Determine the block size by subtracting the decimal value in the subnet's interesting octet from 256.

Step 3. Determine the first subnet by setting all the borrowed bits (which are bits in the subnet mask beyond the bits in the classful subnet mask) to 0.

Step 4. Determine additional subnets by taking the first subnet and counting by the block size increment in the interesting octet.

To reinforce this procedure, consider another example. Say that a 27-bit subnet mask is applied to the network address 192.168.10.0/24. To calculate the created subnets, you can perform the following steps:

Step 1. The subnet mask /27 (in binary) is 11111111.11111111.11111111.11100000. The interesting octet is the fourth octet because the fourth octet contains the last 1 in the subnet mask.

Step 2. The decimal value of the fourth octet in the subnet mask is 224 (11100000 in decimal). Therefore, the block size is 32 (256 – 224 = 32).

Step 3. The first subnet is 192.168.10.0/27—the value of the original 192.168.10.0 network with the borrowed bits (the first 3 bits in the fourth octet) set to 0.

Step 4. Counting by 32 (the block size) in the interesting octet (the fourth octet) allows you to calculate the remaining subnets:

192.168.10.0

192.168.10.32

192.168.10.64

192.168.10.96

192.168.10.128

192.168.10.160

192.168.10.192

192.168.10.224

Now that you know the subnets created from a classful mask given a subnet mask, the next logical step is to determine the usable addresses within those subnets.

Recall that you cannot assign an IP address to a device if all the host bits in the IP address are set to 0 because an IP address with all host bits set to 0 is the address of the subnet itself. Similarly, you cannot assign an IP address to a device if all the host bits in the IP address are set to 1 because an IP address with all host bits set to 1 is the directed broadcast address of a subnet. By excluding the network and directed broadcast addresses from the 192.168.10.0/27 subnets (as previously calculated), you can determine the usable addresses shown in Table 4-26.

<key_topic>

To help develop your subnet-calculation skills, you are now challenged with a few practice subnetting exercises.

Advanced Subnetting Practice: Exercise 1

Using a separate sheet of paper, solve the following scenario:

Based on your network design requirements, you determine that you should use a 26-bit subnet mask applied to your 192.168.0.0/24 network. You now need to calculate each of the created subnets. In addition, you want to know the broadcast address and the range of usable addresses for each of the created subnets.

Advanced Subnetting Practice: Exercise 1 Solution

As described earlier, you can go through the following four-step process to determine the subnet address:

Step 1. The subnet mask /26 (in binary) is 11111111.11111111.11111111.11000000. The interesting octet is the fourth octet because the fourth octet contains the last 1 in the subnet mask.

Step 2. The decimal value of the fourth octet in the subnet mask is 192 (11000000 in decimal). Therefore, the block size is 64 (256 – 192 = 64).

Step 3. The first subnet is 192.168.0.0/26—the value of the original 192.168.0.0 network with the borrowed bits (the first 2 bits in the last octet) set to 0.

Step 4. Counting by 64 (the block size) in the interesting octet (the fourth octet) allows you to calculate the remaining subnets, resulting in the following subnets:

192.168.0.0

192.168.0.64

192.168.0.128

192.168.0.192

The directed broadcast address for each of these preceding subnets can be calculated by adding 63 (that is, one less than the block size) to the interesting octet for each subnet address. Excluding the subnet addresses and directed broadcast addresses, you can calculate a range of usable addresses, the results of which are shown in Table 4-27.

Table 4-27 Usable IP Address Ranges for the 192.168.0.0/26 Subnets

Subnet Address	Directed Broadcast Address	Usable IP Addresses
192.168.0.0	192.168.0.63	192.168.0.1–192.168.0.62
192.168.0.64	192.168.0.127	192.168.0.65– 192.168.0.126
192.168.0.128	192.168.0.191	192.168.0.129– 192.168.0.190
192.168.0.192	192.168.0.255	192.168.0.193– 192.168.0.254

Advanced Subnetting Exercise: Exercise 2

Using a separate sheet of paper, solve the following scenario:

In the network shown in Figure 4-18, the 172.16.0.0/16 network is subnetted using a 20-bit subnet mask. Notice that two VLANs (two subnets) are configured; however, one of the client PCs is assigned an IP address that is not in that PC's VLAN. Which client PC is assigned an incorrect IP address?

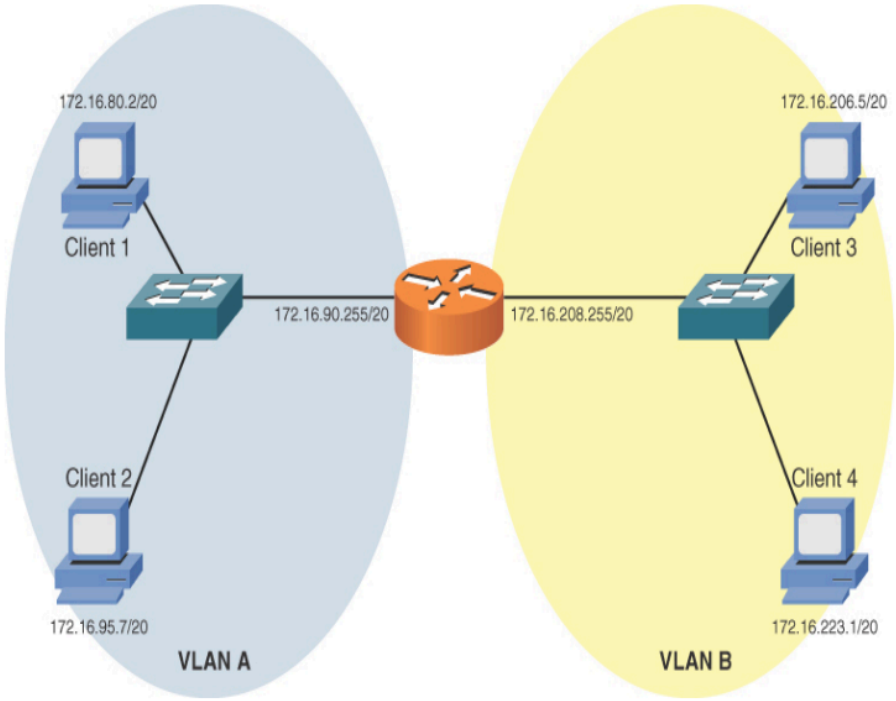


FIGURE 4-18 Topology for Advanced Subnetting Practice: Exercise 2

Advanced Subnetting Practice: Exercise 2 Solution

To determine which client PC is assigned an IP address outside its local VLAN, you need to determine the subnets created by the 20-bit subnet mask applied to the 172.16.0.0/16 network:

Step 1. The interesting octet for a 20-bit subnet mask is the third octet because the third octet is the last octet to contain a 1 in the 20-bit subnet mask (11111111.11111111.11110000.00000000, which could also be written as 255.255.240.0).

Step 2. The decimal value of the third octet in the subnet mask is 240. Therefore, the block size is 16 (256 – 240 = 16).

Step 3. The first 172.16.0.0/20 subnet is 172.16.0.0 (172.16.0.0/20 with the 4 borrowed bits in the third octet set to 0).

Step 4. Beginning with the first subnet, 172.16.0.0/20, and counting by the block size 16 in the interesting octet yields the following subnets:

172.16.0.0/20
172.16.16.0/20
172.16.32.0/20
172.16.48.0/20
172.16.64.0/20
172.16.80.0/20
172.16.96.0/20
172.16.112.0/20
172.16.128.0/20
172.16.144.0/20
172.16.160.0/20
172.16.176.0/20
172.16.192.0/20
172.16.208.0/20
172.16.224.0/20
172.16.240.0/20

Based on the IP addresses of the router interfaces, you can figure out the subnets for VLAN A and VLAN B. Specifically, the router interface in VLAN A has the IP address 172.16.90.255/20. Based on the previous listing of subnets, you can determine that this interface resides in the 172.16.80.0/20 network, whose range of usable addresses is 172.16.80.1–172.16.95.254. Then you can examine the IP addresses of Client 1 and Client 2 to determine whether their IP addresses reside in that range of usable addresses.

Similarly, for VLAN B, the router's interface has an IP address of 172.16.208.255/20. Based on the previous subnet listing, you notice that this interface has an IP address that is part of the 172.16.208.0/20 subnet. As you did for VLAN A, you can check the IP address of Client 3 and Client 4 to decide whether their IP addresses live in VLAN B's range of usable IP addresses (that is, 172.16.208.1–172.16.223.254). Table 4-28 shows these comparisons.

Table 4-28 IP Address Comparison for Advanced Subnetting Practice: Exercise 2

Client	VLAN	Range of Usable Addresses	Client IP Address	Is Client in Range of Usable Addresses?
Client 1	A	172.16.80.1–172.16.95.254	172.16.80.2	Yes
Client 2	A	172.16.80.1–172.16.95.254	172.16.95.7	Yes
Client 3	B	172.16.208.1–172.16.223.254	172.16.206.5	No
Client 4	B	172.16.208.1–172.16.223.254	172.16.223.1	Yes

The comparison in Table 4-28 reveals that Client 3 (with IP address 172.16.206.5) does not have an IP address in VLAN B's subnet (with the usable address range 172.16.208.1–172.16.223.254).

Additional Practice

If you want to continue practicing these concepts, make up your own subnet mask and apply it to a classful network of your choosing. Then you can calculate the created subnets, the directed broadcast IP address for each subnet, and the range of usable IP addresses for each subnet.

To check your work, you can use a subnet calculator. An example of such a calculator is the free Advanced Subnet Calculator, available for download |||||||||||||||||||||||||||||||||||||||||| from https://www.solarwinds.com/free-tools/advanced-subnet-calculator, as shown in Figure 4-19.

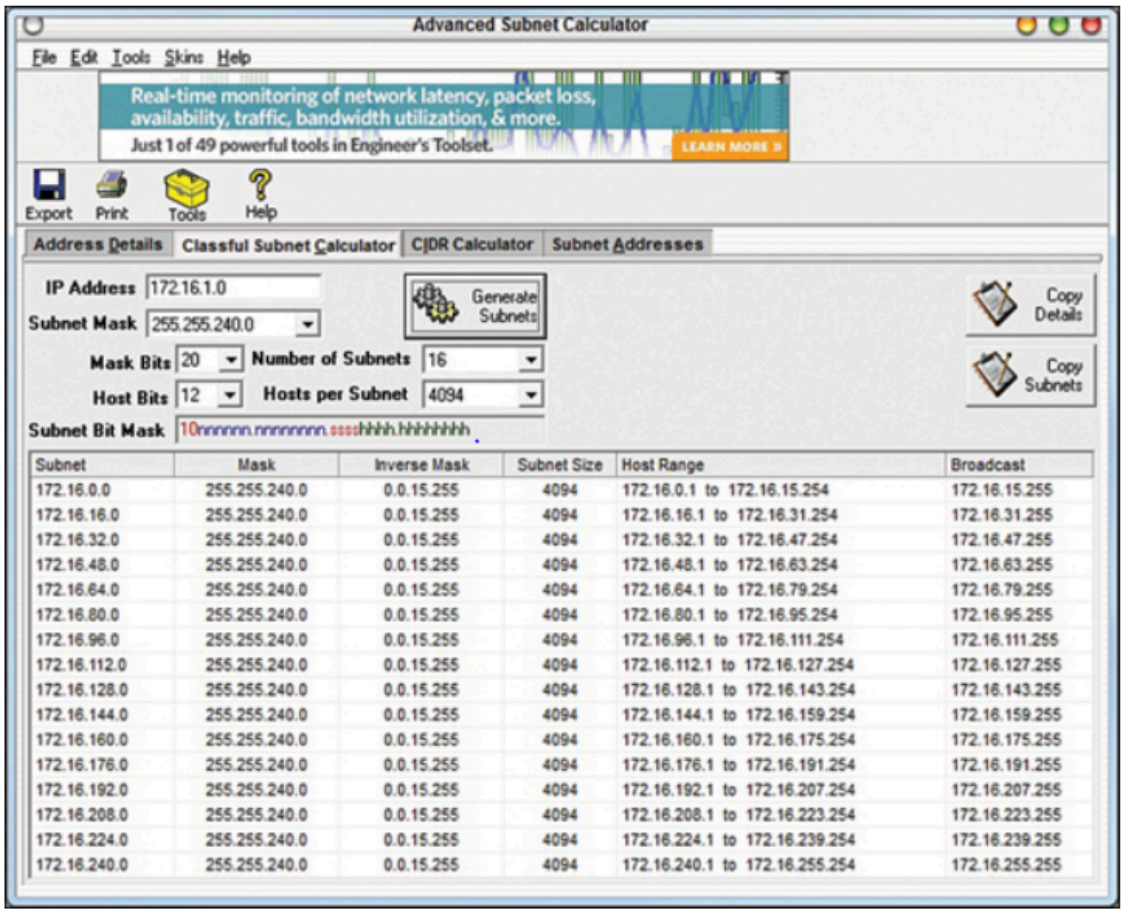


FIGURE 4-19 Free IP Address Manager

Note

As you read through different networking literature, you might come across other approaches to performing subnetting. Various shortcuts exist (including the one presented in this chapter), and some approaches involve much more binary math. The purpose of this section is not to provide an exhaustive treatment of all available subnetting methods but to show a quick and easy approach to performing subnet calculations in the real world and for the Network+ certification exam.

Classless Interdomain Routing

Whereas subnetting is the process of extending a classful subnet mask (that is, adding 1s to a classful mask), *classless interdomain routing* does just the opposite. Specifically, CIDR shortens a classful subnet mask by removing 1s from the classful mask. As a result, CIDR allows contiguous classful networks to be aggregated. This process is sometimes called *route aggregation*.

A typical use of CIDR by a service provider summarizing multiple Class C networks that are assigned to the provider's various customers. For example, imagine that a service provider is responsible for servicing the following Class C networks:

192.168.32.0/24 192.168.33.0/24 192.168.34.0/24 192.168.35.0/24

The service provider could advertise all four networks with the single route advertisement 192.168.32.0/22. To calculate this advertisement, convert the values in the third octet (that is, the octet where the values start to differ) to binary, as shown in Figure 4-20. Then determine how many bits the networks have in common. The number of common bits then becomes the number of bits in the CIDR mask.

Network Address	1st Octet	2nd Octet	3rd Octet	4th Octet
192.168.32.0	11000000	10101000	001000 00	00000000
192.168.33.0	11000000	10101000	001000 01	00000000
192.168.34.0	11000000	10101000	001000 10	00000000
192.168.35.0	11000000	10101000	001000 11	00000000

All Networks Have 22 Bits in Common

<key_topic>

FIGURE 4-20 CIDR Calculation Example

Because all four of the network addresses have the first 22 bits in common, and because setting the remaining bits to 0 (11000000.10101000.00100000.00000000) creates the network address 192.168.32.0, these networks can be summarized as 192.168.32.0/22.

I'd like to acknowledge the many people who contributed their talents to make this book possible: To Tim Green, my acquisitions editor at McGraw Hill: Your encouragement and support during our pandemic edition kept the sanity in place. Love working with you! To my Series Editor, Mike Meyers: I couldn't have done it without you, amigo. Truthfully, has there ever been a better combo than a wizard and a paladin? To Jonathan S. Weissman, technical editor: Great fun working with you on another book. Thanks for keeping me on my toes and technically on point! To Bill McManus, copy editor: I would say that "people say" you're the best in the business, but that people is me. Love working with you! To Travis Everett, writer and editor: Hand in glove on this one, Travis. Love your words and the meticulous attention to nuances that

I missed. Great working with you and look forward to more. Qatar next winter? To Michael Smyer, technologist and photographer: Enjoyed the process on this book, my friend. Yes, even the arguments, because they made the final product much better than anything I could have done on my own. To Dave Rush, senior instructor and top researcher: Thank you for everything you contributed to this book, from research to sounding board to more research . . . . No idea how you can know so much about so much, but I'm very happy you're on my team! To Andrew Hutz, security specialist and wordsmith: Awesome having you on board for this project! Great writing and editing! I look forward to many more with you. To Dudley Lehmer, CEO of Total Seminars: Thanks for keeping the ship afloat while I got to play on this book! You are awesome. To Emily Walters, acquisitions coordinator at McGraw Hill: What a joy to share this project with you! Thanks for keeping us moving and filling in pieces. Good luck with that epic cat and…surf's up! To Rachel Fogelberg, project editor: So fun to work with you! (I hope I didn't add any gray hair with my chronic lateness.) Let's do another one soon. To Janet Walden, editorial supervisor: Thanks for jumping in and donning several hats during vacations and the like. Always enjoy working with you! To Paul Tyler, proofreader: You picked up some great stuff, amigo. This was a new process for us where we got to see your edits and suggestions before we did our own proof. Wow! Thank you. To the KGL compositors and illustrators: The layout was excellent, thanks! And thanks for pushing through at crunch time, too.

By picking up this book, you've shown an interest in learning about networking. But be forewarned: The term *networking* describes a vast field of study, far too large for any certification, book, or training course to cover. Do you want to configure routers and switches for a living? Do you want to administer a large Windows network at a company? Do you want to install wide area network connections? Do you want to set up Web servers? Do you want to secure networks against attacks? If you're considering a CompTIA Network+ certification, you probably don't yet know exactly what aspect of networking you want to pursue, and that's okay! You're going to *love* preparing for the CompTIA Network+ certification exam. Attaining CompTIA Network+ certification provides you with four fantastic benefits. First, you get a superb overview of networking that helps you decide what part of the industry you'd like to pursue. Second, it acts as a prerequisite toward other, more advanced certifications. Third, the amount of eye-opening information you'll gain just makes getting CompTIA Network+ certified plain old *fun*. inally, you'll significantly enhance your opportunity to get a job. Everything is networked today, putting network techs in demand. Nothing comes close to providing a better overview of networking than CompTIA Network+. The certification covers local area networks (LANs), wide area networks (WANs), the Internet (the world's largest WAN), security, cabling, and applications in a wide-but-not-too-deep fashion that showcases the many different parts of a network and hopefully tempts you to investigate the aspects that intrigue you by looking into follow-up certifications. The process of attaining CompTIA Network+ certification will give you a solid foundation in the whole field of networking. Mastering the competencies will help fill in gaps in your knowledge and provide an ongoing series of "a-ha!" moments of grasping the big picture that make being a tech so much fun. Ready to learn a lot, grab a great certification, and have fun doing it? Then welcome to CompTIA Network+ certification!

Who Needs CompTIA Network+? I Just Want to Learn About Networks!

Whoa there, amigo! Are you one of those folks who either has never heard of the CompTIA Network+ exam or just doesn't have any real interest in certification? Is your goal only to get a solid handle on networks and a jump start on the basics? Are you looking for that "magic bullet" book that you can read from beginning to end and then start installing and troubleshooting a network? Do you want to know what's involved with running network cabling in your walls or getting your new wireless network working? Are you tired of not knowing TCP/IP and how it works? If these types of questions are running through your mind, then rest easy – you have the right book. Like every book with the Mike Meyers name, you'll get solid concepts without pedantic details or broad, meaningless overviews. You'll look at real-world networking as performed by real techs. This is a book that understands your needs and goes well beyond the scope of a single certification. If the CompTIA Network+ exam isn't for you, you can skip the rest of this introduction, shift your brain into learn mode, and dive into Chapter 1. But then, if you're going to have the knowledge, why *not* get the certification?

What is CompTIA Network+ Certification?

CompTIA Network+ certification is an industry-wide, vendor-neutral certification program developed and sponsored by the Computing Technology Industry Association (CompTIA). The CompTIA Network+ certification shows that you have a basic competency in the physical support of networking systems and knowledge of the conceptual aspects of networking. To date, many hundreds of thousands of technicians have become CompTIA Network+ certified. CompTIA Network+ certification enjoys wide recognition throughout the IT industry. It is considered the obvious next step after CompTIA A+ certification. (CompTIA A+ is the certification for PC technicians.)

The Current CompTIA Network+ Certification Release

CompTIA constantly works to provide exams that cover the latest technologies and, as part of that effort, periodically updates its certification objectives, domains, and exam questions. This book covers all you need to know to pass the N10-008 CompTIA Network+ exam released in 2021.

How Do I Become CompTIA Network+ Certified?

To become CompTIA Network+ certified, you must simply pass one computer-based exam. There are no prerequisites for taking the CompTIA Network+ exam, and no networking experience is needed. You're not required to take a training course or buy any training materials. The only requirements are that you pay a testing fee to an authorized testing facility and then sit for the exam. Upon completion of the exam, you will immediately know whether you passed or failed. Once you pass, you become CompTIA Network+ certified for three years. After three years, you'll need to renew your certification by taking the current exam or completing approved Continuing Education activities. By completing these activities, you earn credits that (along with an annual fee) allow you to keep your CompTIA Network+ certification. For a full list of approved

activities, check out CompTIA's Web site (www.comptia .org) and search for **CompTIA Continuing Education Program**.

NOTE The American National Standards Institute (ANSI) has accredited the CompTIA Network+ certification as compliant with the ISO/IEC 17024 standard. That makes it special.

Now for the details: CompTIA recommends that you have at least nine to twelve months of networking experience and CompTIA A+ knowledge, but this is not a requirement. Note the word "recommends." You may not need experience or CompTIA A+ knowledge, but each helps! The CompTIA A+ certification competencies have a degree of overlap with the CompTIA Network+ competencies, such as types of connectors and how networks work. As for experience, keep in mind that CompTIA Network+ is mostly a practical exam. Those who have been out there supporting real networks will find many of the questions reminiscent of the types of problems they have seen on LANs. The bottom line is that you'll probably have a much easier time on the CompTIA Network+ exam if you have some CompTIA A+ experience under your belt.

The CompTIA Network+ exam is extremely practical. Questions often present real-life scenarios and you present the best solution. The CompTIA Network+ exam loves troubleshooting. Let me repeat: many of the test objectives deal with direct, *real-world troubleshooting*. Be prepared to troubleshoot both hardware and software failures and to answer both "What do you do next?" and "What is most likely the problem?" types of questions. A qualified CompTIA Network+ certification candidate can install and configure a PC to connect to a network. This includes installing and testing a network card, configuring drivers, and loading all network software. The exam will test you on the different topologies, standards, and cabling. Expect conceptual questions about the Open Systems Interconnection (OSI) sevenlayer model. You need to know the functions and protocols for each layer to pass the CompTIA Network+ exam. You can also expect questions on most of the protocol suites, with heavy emphasis on the TCP/IP suite. If you've never heard of the OSI seven-layer model, don't worry! This book will teach you all you need to know.

NOTE CompTIA occasionally makes changes to the content of the exam, as well as the score necessary to pass it. Always check the Web site of my company, Total Seminars (www.totalsem.com), before scheduling your exam.

How Do I Take The Test?

To take the test, you may go to an authorized testing center or take it over the Internet. Pearson VUE administers the actual CompTIA Network+ exam. You'll find thousands of Pearson VUE testing centers scattered across the United States and Canada, as well as in over 186 other countries around the world. You may take the exam at any testing center. To locate a testing center and schedule an exam, call Pearson VUE at 877-551-7587. You can also visit their Web site at https://home.pearsonvue.com. To schedule an Internetbased exam through OnVUE, go

to www.onvue.com. You'll need a solid Internet connection and a webcam, such as one built into most portable computers.

How Much Does the Test Cost?

CompTIA fixes the price, no matter what testing center you use. The cost of the exam depends on whether you work for a CompTIA member. At press time, the cost for nonCompTIA members is $338 (U.S.). If your employer is a CompTIA member, you can save money by obtaining an exam voucher. In fact, even if you don't work for a CompTIA member, you can purchase a voucher from member companies (like mine) and take advantage of significant member savings. You simply buy the voucher and then use the voucher to pay for the exam. Vouchers are delivered to you on paper and electronically via e-mail. The voucher number is the important thing. That number is your exam payment, so protect it from fellow students until you're ready to schedule your exam. If you're in the United States or Canada, you can visit www.totalsem.com or call 800- 446-6004 to purchase vouchers. As I always say, "You don't have to buy your voucher from us, but for goodness' sake, get one from somebody!" Why pay full price when you have a discount alternative? You must pay for the exam when you schedule, whether online or by phone. If you're scheduling by phone, be prepared to hold for a while. Have your Social Security number (or the international equivalent) ready and either a credit card or a voucher number when you call or begin the online scheduling process. If you require any special accommodations, Pearson VUE will be able to assist you, although your selection of testing locations may be a bit more limited. International prices vary; see the CompTIA Web site for international pricing. Of course, prices are subject to change without notice, so always check the CompTIA Web site for current pricing!

Obligate Yourself

The first step you should take is to schedule the exam. Ever heard the old adage that heat and pressure make diamonds? Well, if you don't give yourself a little "heat," you might procrastinate and unnecessarily delay taking the exam. Even worse, you may end up not taking the exam at all. Do yourself a favor. Determine how much time you need to study CompTIA Network+ Certification All-in-One Exam Guide xxx (see the next section), and then call Pearson VUE and schedule the exam, giving yourself the time you need to study—and adding a few extra days for safety. Afterward, sit back and let your anxieties wash over you. Suddenly, turning off the smartphone and cracking open the book will become a lot easier.

Set Aside the Right Amount of Study Time

After helping thousands of techs get their CompTIA Network+ certification, we at Total Seminars have developed a pretty good feel for the amount of study time needed to pass the CompTIA Network+ exam. Table 1 will help you plan how much study time you must devote to the exam. Keep in mind that these are averages. If you're not a great student or if you're a little on the nervous side, add another 10 percent. Equally, if you're the type who can learn an entire semester of geometry in one night, reduce the numbers by 10 percent. To use this table, just

circle the values that are most accurate for you and add them up to get the number of study hours.

Network Models - Chapter 1

The CompTIA Network+ certification exam expects you to know how to:

*1.1 Compare and contrast the Open Systems Interconnection (OSI) model layers and encapsulation concepts
*1.2 Explain the characteristics of network topologies and network types
*2.1 Compare and contrast various devices, their features, and their appropriate placement on the network

To achieve these goals, you must be able to:

* Describe how the OSI seven-layer model helps technicians understand and troubleshoot networks
* Explain the major functions of networks with the OSI seven-layer model

Networks enabled connected hosts – computers – to share resources and access resources. The sharing host – a server – runs special software to enable the accessing host – a client – to get the desired resource. That resource can be any number of things, from a Web page on the Internet to files on a server in your office. It might even be a printer or a camera. Networking professionals need to know how all the connections happen and all the hardware and software that enables that exchange of resources.

The CompTIA Network+ certification challenges you to understand virtually every aspect of networking—not a small task. Networking professionals use models to conceptualize the many parts of a network, relying primarily on the *Open Systems Interconnect (OSI) seven-layer model*. The OSI model provides two tools that makes it essential for networking techs. First, the OSI model provides a powerful mental tool for diagnosing problems. Understanding OSI enables a tech to determine quickly at what layer a problem can occur and helps the tech zero in on a solution without wasting a lot of time on false leads. Second, the OSI model provides a common language techs use to describe specific network functions. Figure 1-1 shows product information for a Cisco-branded advanced networking device. Note the use of the terms “L3” and “layer 7.” These terms directly reference the OSI seven-layer model. Techs who understand the OSI model understand what those numbers mean, giving them a quick understanding of what the device provides to a network.

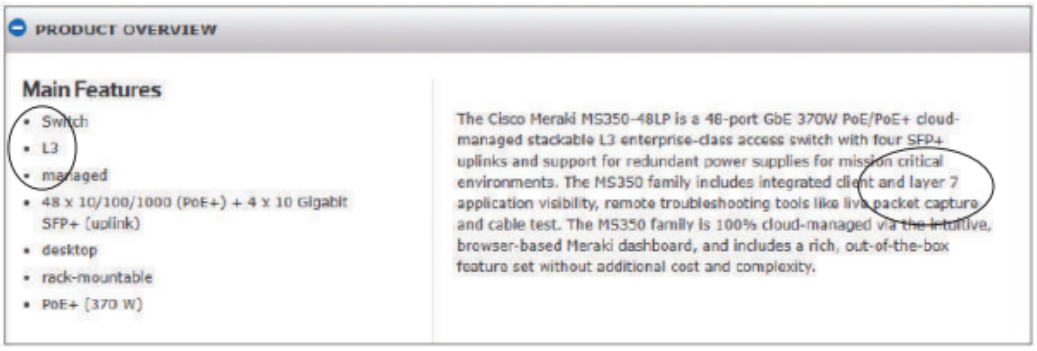


Figure 1-1 Using OSI terminology in device documentation

This chapter looks first at models in general and how models help conceptualize and troubleshoot networks. The chapter then explores the OSI seven-layer model to see how it helps clarify network architecture for techs.

Historical/Conceptual

Networking is hard. It takes a lot of pieces, both hardware and software, all working incredibly quickly and in perfect harmony, to get anything done. Just making Google appear in your Web browser requires millions of hours in research, development, and manufacturing to create the many pieces to successfully connect your system to a server somewhere in Googleland and to enable them to communicate. Whenever we encounter highly complex technologies, we need to simplify the overall process by breaking it into discrete, simple, individual processes. We do this using a network *model*.

Biography of a Model

What does the word “model” mean to you? Does the word make you think of a person walking down a catwalk at a fashion show in some outrageous costume or another showing off the latest style of blue jeans on a huge billboard? Maybe it makes you think of a plastic model airplane? What about those computer models that try to predict weather? We use the term “model” in a number of ways, but each use shares certain common themes. All models are a simplified representation of the real thing. The human model ignores the many different types of body shapes, using only a single “optimal” figure. The model airplane lacks functional features or the internal framework, and the computerized weather model might disregard subtle differences in wind temperatures or geology (Figure 1-2).

Figure 1-2 Types of models

Additionally, a model must have at least all the major functions of the real item, but what constitutes a major rather than a minor function is open to opinion. Figure 1-3 shows a different level of detail for a model. Does it contain all the major components of an airplane? There's room for argument that perhaps the model should have landing gear to go along with the propeller, wings, and tail.

Figure 1-3
Simple model airplane

Network Models

Network models face similar challenges. What functions define all networks? What details can you omit without rendering the model inaccurate? Does the model retain its usefulness when describing a network that does not omit all the layers?

In the early days of networking, different manufacturers made unique types of networks that functioned well. Part of the reason the networks worked was that every network manufacturer made everything. Back then, a single manufacturer provided everything for a customer when the customer purchased a network solution: all the hardware and all the software in one complete and expensive package. Although these networks worked fine as stand-alone networks, the proprietary nature of the hardware and software made it difficult—to put it

mildly—to connect networks of multiple manufacturers. To interconnect networks and therefore improve the networking industry, someone needed to create a guide, a model, that described the functions of a network. Using this model, the people who made hardware and software could work together to make networks that worked together well.

NOTE The International Organization for Standardization (ISO) created the OSI seven-layer model. SO may look like a misspelled acronym, but it's actually a word, derived from the Greek word *isos*, which means "equal". The International Organization for Standardization sets standards that promote *equality* among network designers and manufacturers, thus ISO.

The best way to learn the OSI model is to see it in action. For this reason, I'll introduce you to a small, simplified network that needs to copy a file from one computer to another. This example goes through each of the OSI layers needed to copy that file, and I explain each step and why it is necessary. The next part of the chapter explores a Web-centric enterprise version of a company so you can see how the OSI model applies to the latest networks. By the end of the chapter, you should have a definite handle on using the OSI model as a tool to conceptualize networks. You'll continue to build on this knowledge throughout the book and turn your OSI model knowledge into a powerful troubleshooting tool.

The OSI Seven-Layer Model on a Simple Network

Each layer in the OSI seven-layer model defines an important function in computer networking, and the protocols that operate at that layer offer solutions to those functions. *Protocols* are sets of clearly defined rules, regulations, standards, and procedures that enable hardware and software developers to make devices and applications that function properly at a particular layer. The OSI seven-layer model encourages modular design in networking, meaning that each layer has as little to do with the operation of other layers as possible. Think of it as an automobile assembly line. The guy painting the car doesn't care about the gal putting doors on the car—he expects the assembly line process to make sure the cars he paints have doors. Each layer on the model trusts that the other layers on the model do their jobs. The OSI seven layers are:

- **Layer 7:** Presentation
- **Layer 6:** Presentation
- **Layer 5:** Session
- **Layer 4:** Transport
- **Layer 3:** Network
- **Layer 2:** Data Link
- **Layer 1:** Physical

The OSI seven layers are not laws of physics—anybody who wants to design a network can do it any way he or she wants. Although many protocols fit neatly into one of the seven layers, others do not.

EXAM TIP: Be sure to memorize both the name and the number of each OSI layer.

Network techs use OSI terms such as “Layer 4” and “Transport layer” synonymously. Students have long used mnemonics for memorizing such lists. One of my favorites for the OSI seven-layer model “Please Do Not Throw Sausage Pizza Away.” Yum! ! Another great mnemonic that helps students to memorize the layers from the top down is “All People Seem To Need Data Processing.” Go with what works for you. Now that you know the names of the layers, let’s see what each layer does. The best way to understand the OSI layers is to see them in action. Let’s see them at work at the fictional company of MHTechEd, Inc.

NOTE: This section is a conceptual overview of the hardware and software functions of a network. Your network may have different hardware or software, but it will share the same functions.

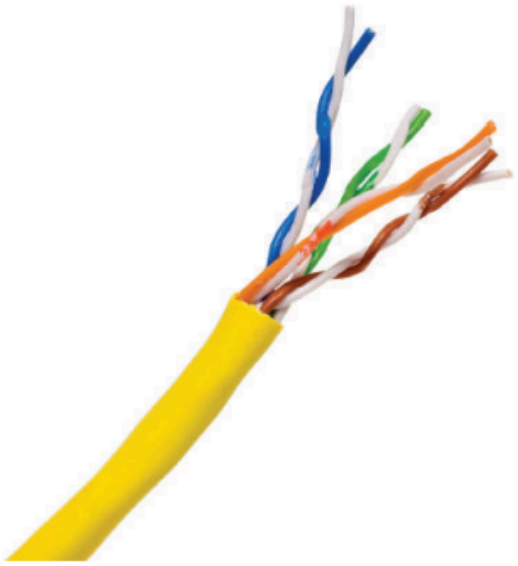
Because of the kinds of work they do, these two often need to exchange data between their two PCs. At the moment, Shannon has just completed a new employee handbook in Microsoft Word, and she wants Scott to check it for accuracy. Shannon could transfer a copy of the file to Scott’s computer by the tried-and-true Sneakernet method—saving the file on a flash drive and walking it over to him—but thanks to the wonders of computer networking, she doesn’t even have to get up from her chair. Let’s watch in detail each piece of the process that gives Scott direct access to Shannon’s computer, so he can copy the Word document from Shannon’s system to her own. Long before Shannon ever saved the Word document on her system—when the systems were first installed—someone who knew what they were doing set up and configured all the systems at MHTechEd to be part of a common network. All this setup activity resulted in multiple layers of hardware and software that can work together behind the scenes to get that Word document from Shannon’s system to Scott’s. Let’s examine the different pieces of the network.

Test Specific

Let’s Get Physical—Network Hardware and Layers 1–2

Clearly the network needs a physical channel through which it can move bits of data between systems. Most networks use a cable like the one shown in Figure 1-5. This cable, known in the networking industry as *unshielded twisted pair*, usually contains four pairs of wires that can transmit and receive data.

Figure 1-5
UTP cabling



Another key piece of hardware the network uses is a special box-like device that handles the flow of data from each computer to every other computer (Figure 1-6). This box is often tucked away in a closet or an equipment room. (The technology of the central box has changed over time. For now, let's just call it the "central box." I'll get to variations in a bit.) Each system on the network has its own cable that runs to the central box.

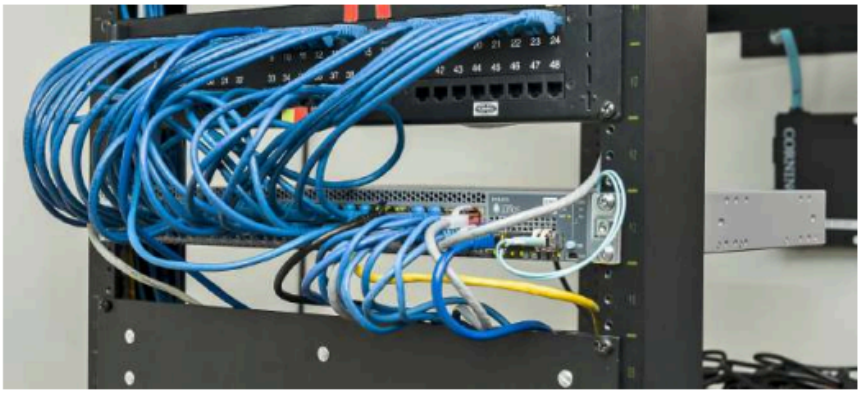


Figure 1-6 Typical central box

Layer 1 of the OSI model defines the method of moving data between computers, so the cabling and central box are part of the Physical layer (Layer 1). Anything that moves data from one system to another, such as copper cabling, fiber optics, even radio waves, is part of the OSI Physical layer. Layer 1 doesn't care what data goes through; it just moves the data from one system to another system. Figure 1-7 shows the MHTechEd network in the OSI seven-layer

model thus far. Note that each system has the full range of layers, so data from Shannon's computer can flow to Scott's computer. (I'll cover what a "hub" is shortly.)

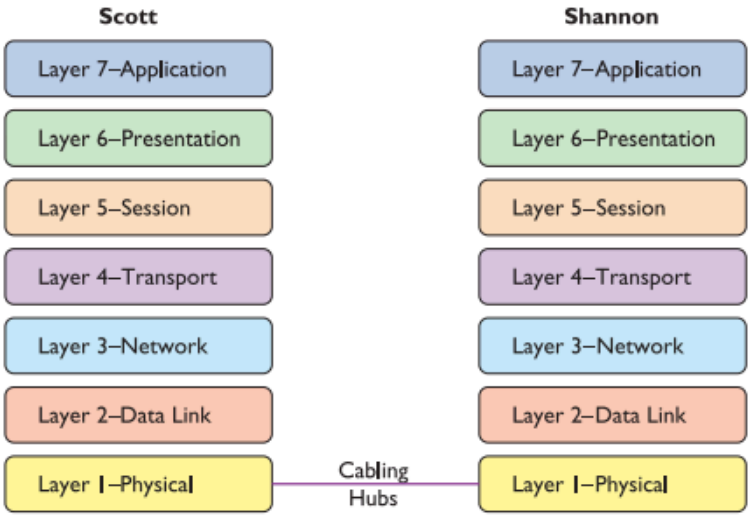


Figure 1-7
The network so far, with the Physical layer hardware installed

The real magic of a network starts with the *network interface card*, or *NIC (*pronounced "nick"), which serves as the interface between the PC and the network. While NICs come in a wide array of shapes and sizes, the ones at MHTechEd look like Figure 1-8.

Figure 1-8
Typical NIC

On older systems, a NIC truly was a separate card that snapped into a handy expansion slot, which is why they were called network interface *cards*. Even though they're now built into the motherboard, they are still called NICs. Figure 1-9 shows a typical modern laptop with a dongle providing an Ethernet port. Note the cable runs from the NIC into the wall; inside that wall is another cable running all the way back to the central box.

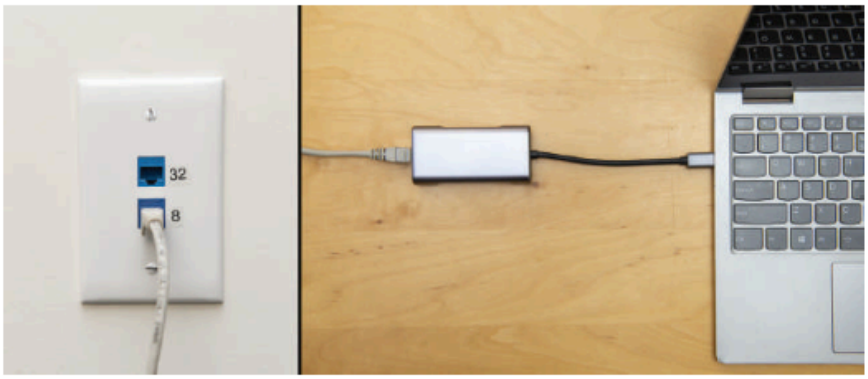
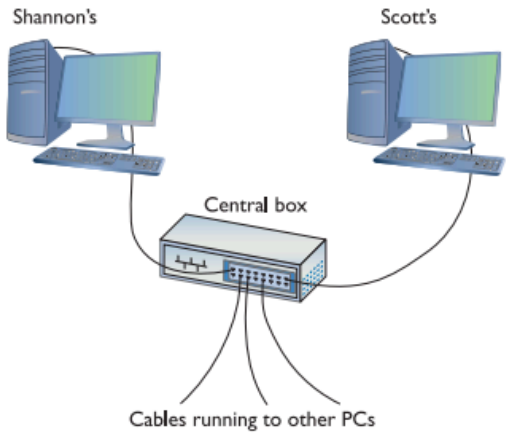


Figure 1-9 Dongle NIC with cable connecting the laptop to the wall jack

Cabling and central boxes define the Physical layer of the network, and NICs provide the interface to the PC. Figure 1-10 shows a diagram of the network cabling system. I'll build on this diagram as I delve deeper into the network process.

Figure 1-10 The MHTechEd network



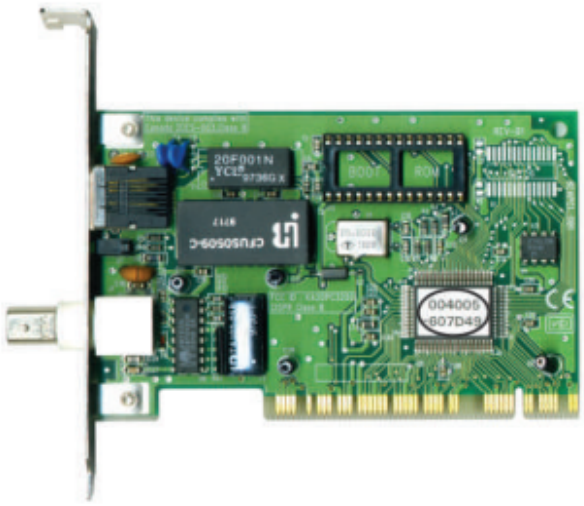
You might be tempted to categorize the NIC as part of the Physical layer at this point, and you'd have a valid argument. The NIC clearly is necessary for the physical connection to take place. Many authors put the NIC in OSI Layer 2, the Data Link layer, though, so clearly something else is happening inside the NIC. Let's take a closer look.

The NIC

To understand how networks work, you must understand how NICs work. The network must provide a mechanism that gives each system a unique identifier—like a telephone number—so data is delivered to the right system. That's one of the NIC's most important jobs. Inside every NIC, burned onto some type of ROM chip, is special firmware containing a unique identifier with a 48-bit value called the *media access control address, or MAC address*. No two NICs ever share the same MAC address – ever. Any company that makes NICs must contact the Institute of Electronics and Electrical Engineering (IEEE) and request a block of MAC addresses, which

the company then burns into the ROMs on its NICs. Many NIC makers also print the MAC address on the surface of each NIC, as shown in Figure 1-11. Note that the NIC shown here displays the MAC address in hexadecimal notation. Count the number of hex characters—because each hex character represents 4 bits, it takes 12 hex characters to represent 48 bits. MAC addresses are always written in hex.

Figure 1-11
MAC address



Hexadecimal Aside

A hexadecimal numbering system uses base 16 to represent numbers—that would be 0–15 (in base 10 values). Contrast this with the more common decimal numbering system, numbered 0–9. Just as with decimal, people who work with hexadecimal need a single character to represent each number for the 16 values. Using 0–9 makes sense, but then hex is represented in letter form for the values 10–15 (A, B, C, D, E, F). Hexadecimal works great with binary. Four bits provide the values of 0–15. 0001, for example, is the value 1; 1000 in binary is 8; 1111 is 15. When we work with MAC addresses, it's far easier to break each 4-bit section of the 48-bit address and translate that into hex. Humans work better that way!

Back to MAC Addresses

The MAC address in Figure 1-11 is 004005-607D49, although in print, we represent the MAC address as 00–40–05–60–7D–49. The first six digits, in this example, 00-40-05, represent the number of the NIC manufacturer. Once the IEEE issues those six hex digits to a manufacturer—referred to as the *Organizationally Unique Identifier* – no other manufacturer may use them. The last six digits, in this example 60–7D–49, are the manufacturer's unique serial number for that NIC; this portion of the MAC is often referred to as the *device ID*.

NOTE Windows uses the dash as the delimiter for the MAC address. Linux and macOS use a colon.

Would you like to see the MAC address for your NIC? If you have a Windows system, type ipconfig /all from a command prompt to display the MAC address (Figure 1-12). Note that ipconfig calls the MAC address the physical address, which is an important distinction, as you'll see a bit later in the chapter. For macOS, type ifconfig from a terminal; for Linux, type ip a from a terminal to get similar results. Figure 1-13 shows a Kali Linux terminal; the link/ether line shows the MAC address.

MAC-48 and EUI-48

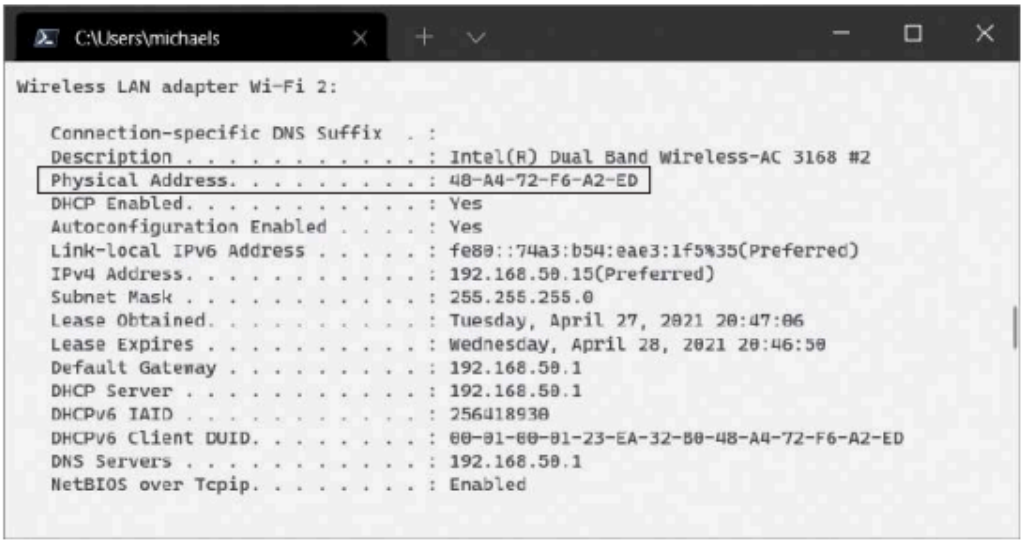


Figure 1-12 Output from `ipconfig /all`

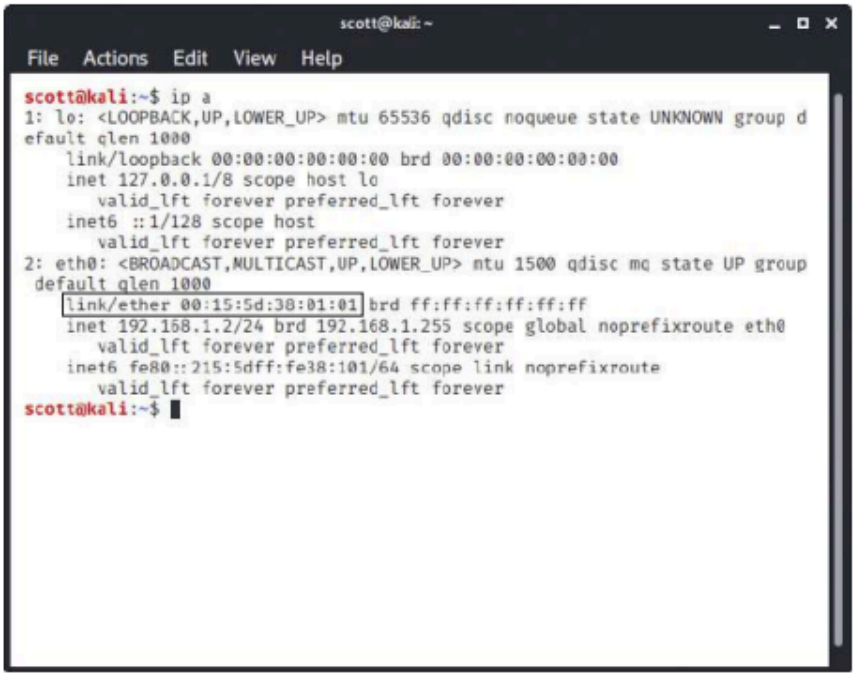


Figure 1-13 Output from `ip a` in Kali Linux

Okay, so every NIC in the world has a unique MAC address, but how is it used? Ah, that's where the fun begins! Recall that computer data is binary, which means it's made up of streams of ones and zeroes. NICs send and receive this binary data as pulses of electricity, light, or radio waves. Let's consider the NICs that use electricity to send and receive data. The specific process by which a NIC uses electricity to send and receive data is exceedingly complicated

but, luckily for you, not necessary to understand. Instead, just think of a *charge* on the wire as a one and *no charge* as a zero. A chunk of data moving in pulses across a wire might look something like Figure 1-14.

Try This!

What's Your MAC Address?

You can readily determine your MAC address on a desktop computer. 1. On macOS systems, open a terminal, type ifconfig, and press the enter key. 2. On Linux systems, open a terminal, type ip a, and press the enter key. 3. In Windows, type cmd at the Start screen and press enter when the Command Prompt option appears on the right. At the command prompt, type the command ipconfig /all and press the enter key.

If you put an oscilloscope on the wire to measure voltage, you'd see something like Figure 1-15. An oscilloscope is a powerful tool that enables you to see electrical pulses.

Figure 1-15
Oscilloscope of data

Now, remembering that the pulses represent binary data, visualize instead a string of ones and zeroes moving across the wire (Figure 1-16).

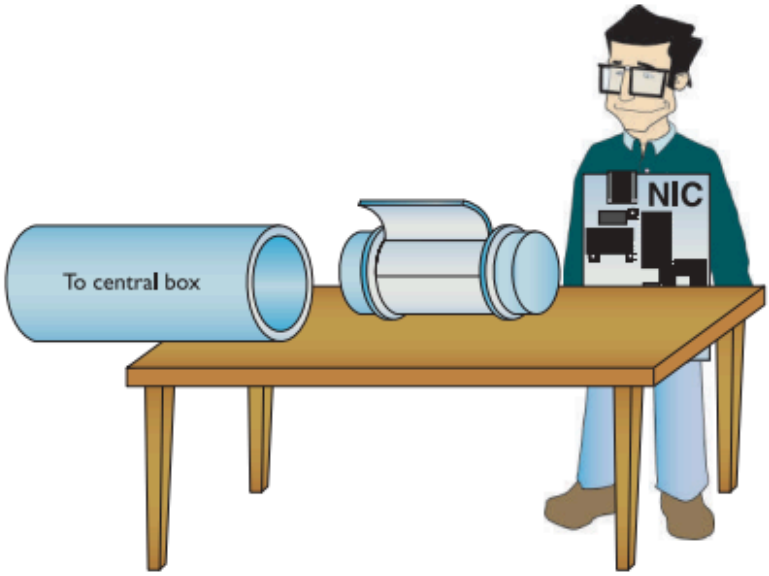
Figure 1-16
Data as ones and zeroes

Once you understand how data moves along the wire, the next question is, how does the network get the right data to the right system? All networks transmit data by breaking whatever is moving across the Physical layer (such as files, print jobs, Web pages, and so forth) into discrete chunks called frames. A *frame* is basically a container for a chunk of data moving across a network. A frame *encapsulates* – puts a wrapper around – information and data for easier transmission. (More on this later in the chapter.) The NIC creates and sends, as well as receives and reads, these frames.

NOTE The unit of data specified by a protocol at each layer of the OSI seven-layer model is called a *protocol data unit*. A frame is the PDU for Layer 2.

them out on the wire to the central box (Figure 1-17).

Figure 1-17
Inside the NIC



Here's where the MAC address becomes important. Figure 1-18 shows a representation of a generic frame, a simplified version of the wired network technology for home/ office use, called *Ethernet*. (Chapter 3 covers Ethernet in great detail. For now just go with the frame here as a generic wired thing.)

Figure 1-18
Generic frame

Even though a frame is a string of ones and zeroes, we often draw frames as a series of rectangles, each rectangle representing a part of the string of ones and zeroes. You will see this type of frame representation used quite often, so you should become comfortable with it (even though I still prefer to see frames as pneumatic canisters). Note that the frame begins with the MAC address of the NIC to which the data is to be sent, followed by the MAC address of the sending NIC. Next comes the *Type* field, which indicates what's encapsulated in the frame. Then comes the *Data* field that contains what's encapsulated, followed by a special piece of checking information called the *frame check sequence (FCS)*. The FCS uses a type of binary math called a *cyclic redundancy check (CRC)* that the receiving NIC uses to verify that the data arrived intact. You can think of a frame in a different way as having three sections. The *header* (MAC addresses and Type) starts, followed by the *payload* (whatever is encapsulated in the frame); this is followed by the *trailer* (the FCS).

to diamonds—the NIC doesn't care one bit (pardon the pun).

Figure 1-19
Frame as a canister

Like a canister, a frame can hold only a certain amount of data. Different types of networks use different sizes of frames, but the frames used in Ethernet networks hold at most 1500 bytes of data. This raises a new question: what happens when the data to be sent is larger than the frame size? Well, the sending system's software must chop the data up into nice, frame-sized chunks, which it then hands to the NIC for sending. As the receiving system begins to accept the incoming frames, the receiving system's software recombines the data chunks as they come in from the network. I'll show how this disassembling and reassembling is done in a moment—first, let's see how the frames get to the right system!

Into the Central Box

When a system sends a frame out on the network, the frame goes into the central box. What happens next depends on the technology of the central box. In the early days of networking, the central box was called a *hub*. A hub was a dumb device, just a repeater. When it received a frame, the hub made an exact copy of that frame, sending a copy of the original frame out of all connected ports except the port on which the message originated.

The interesting part of this process was when the copy of the frame came into all the other systems. I like to visualize that a frame slid onto the receiving NIC's "frame assembly table," where the electronics of the NIC inspected it. (This doesn't exist; use your imagination!) Here's where the magic took place: only the NIC to which the frame was addressed would process that frame—the other NICs simply dropped it when they saw that it was not addressed to their MAC address. This is important to appreciate: with a hub, *every* frame sent on a network was received by *every* NIC, but only the NIC with the matching MAC address would process that frame (Figure 1-20).

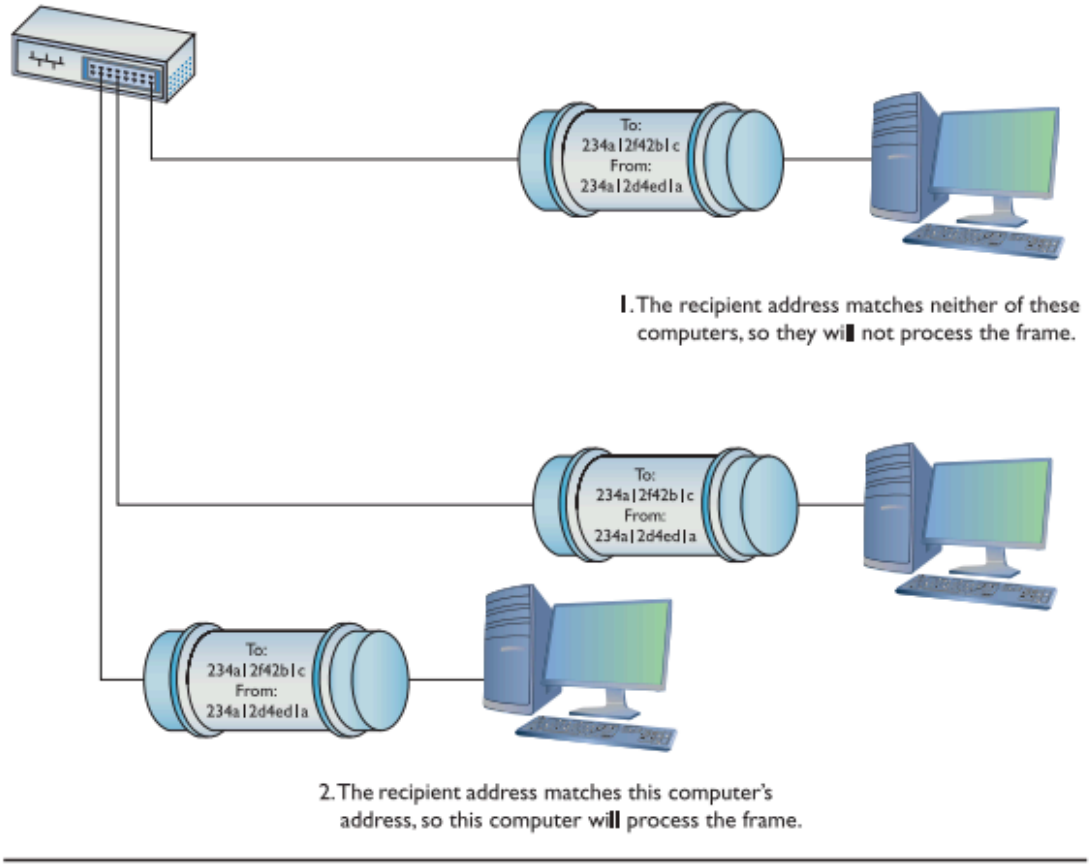


Figure 1-20 Incoming frame!

Later networks replaced the hub with a smarter device called a *switch*. Switches, as you'll see in much more detail as we go deeper into networking, filter traffic by MAC address. Rather than sending all incoming frames to all network devices connected to it, a switch sends the frame only to the interface associated with the destination MAC address.

FCS in Depth

All FCSs are only 4 bytes long, yet the wired frame carries at most 1500 bytes of data. How can 4 bytes tell you if all 1500 bytes in the data are correct? That's the magic of the math of the CRC. Without going into the grinding details, think of the CRC as just the remainder of a division problem. (Remember learning remainders from division back in elementary school?) The NIC sending the frame does a little math to make the CRC. The receiving NIC applies the same math. If the receiving NIC's answer is the same as the CRC, it knows the data is good; if it's not good, the frame is dropped.

Getting the Data on the Line

The process of getting data onto the wire and then picking that data off the wire is amazingly complicated. For instance, what would happen to keep two NICs from speaking at the same

time? Because all the data sent by one NIC is read by every other NIC on the network, only one system could speak at a time in early wired networks. Networks use frames to restrict the amount of data a NIC can send at once, giving all NICs a chance to send data over the network in a reasonable span of time. Dealing with this and many other issues requires sophisticated electronics, but the NICs handle these issues completely on their own without our help. Thankfully, the folks who design NICs worry about all these details, so we don't have to!

Getting to Know You

Using the MAC address is a great way to move data around, but this process raises an important question. How does a sending NIC know the MAC address of the NIC to which it's sending the data? In most cases, the sending system already knows the destination MAC address because the NICs had probably communicated earlier, and each system stores that data. If it doesn't already know the MAC address, a NIC may send a *broadcast* on that network to ask for it. The MAC address of FF-FF-FF-FF-FF-FF is the Layer 2 *broadcast address* – if a NIC sends a frame using the broadcast address, every single NIC on the network will process that frame. That broadcast frame's data will contain a request for a system's MAC address. Without knowing the MAC address to begin with, the requesting computer will use an IP address to pick the target computer out of the crowd. The system with the MAC address your system is seeking will read the request in the broadcast frame and respond with its MAC address. (See "IP—Playing on Layer 3, the Network Layer" later in this chapter for more on IP addresses and packets.)

The Complete Frame Movement

Now that you've seen all the pieces needed to send and receive frames, let's put these pieces together and see how a frame gets from one system to another. The basic send/receive process is as follows. First, the sending system's operating system hands some data to its NIC. The NIC builds a frame to transport that data to the receiving NIC (Figure 1-21).

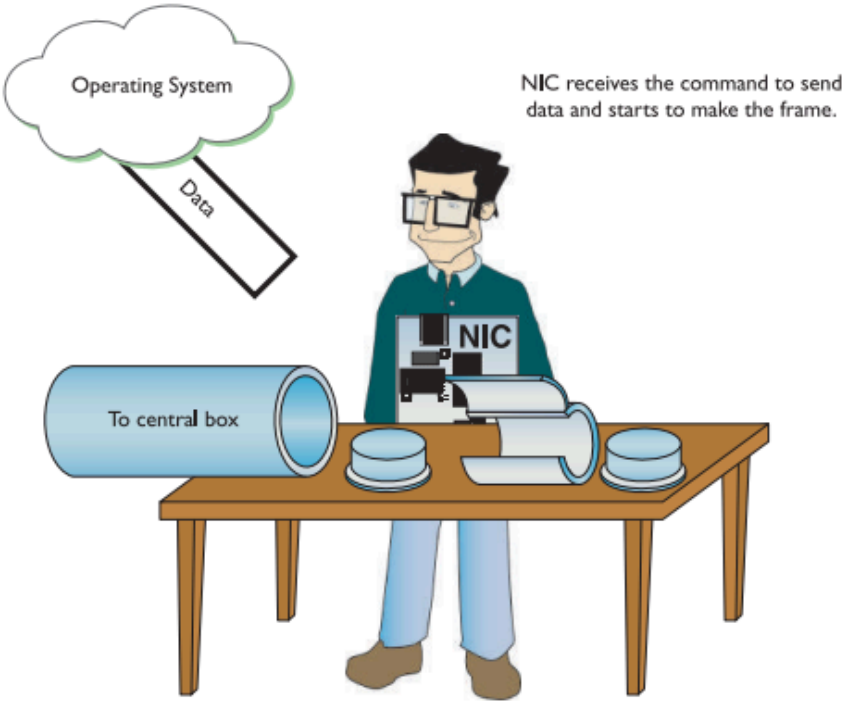


Figure 1-21 Building the frame

After the NIC creates the frame, it adds the FCS, and then dumps it and the data into the frame (Figure 1-22).

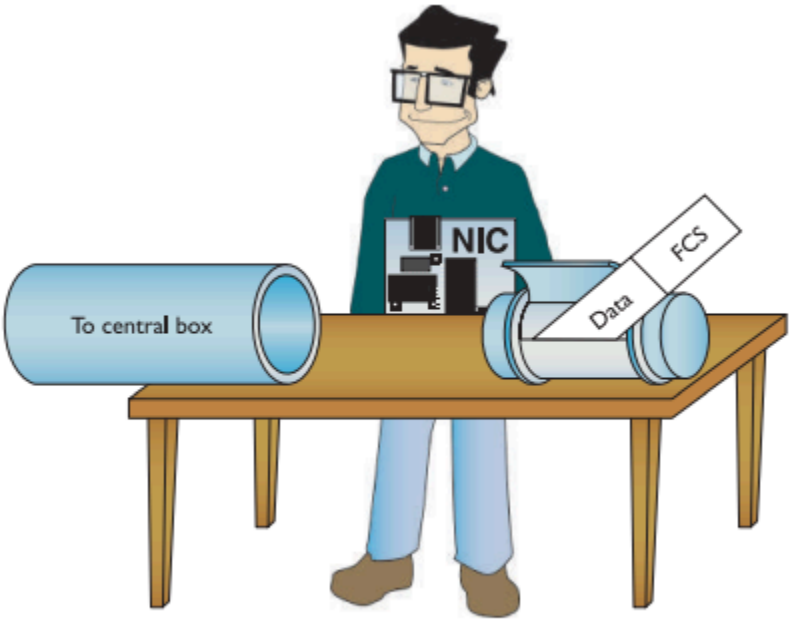


Figure 1-22 Adding the data and FCS to the frame

Next, the NIC puts both the destination MAC address and its own MAC address onto the frame. It then sends the frame through the cable to the network (Figure 1-23).

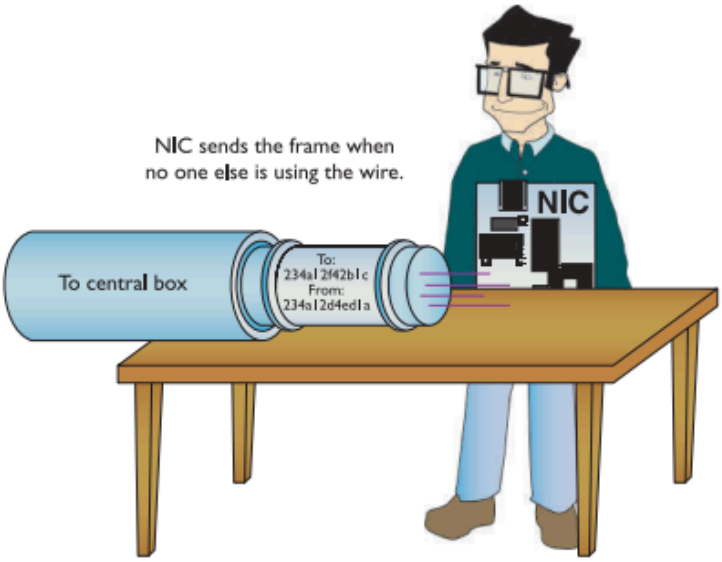


Figure 1-23 Sending the frame

The frame propagates down the wire into the central box. The switch sends unicast frames to the destination address and sends broadcast frames to every system on the network. The NIC receives the frame (Figure 1-24). The NIC strips off all the framing information and sends the data to the software—the operating system—for processing. The receiving NIC doesn't care what the software does with the data; its job stops the moment it passes on the data to the software. Any device that deals with a MAC address is part of the OSI *Data Link layer*, or Layer 2 of the OSI model. Let's update the OSI model to include details about the Data Link layer (Figure 1-25).

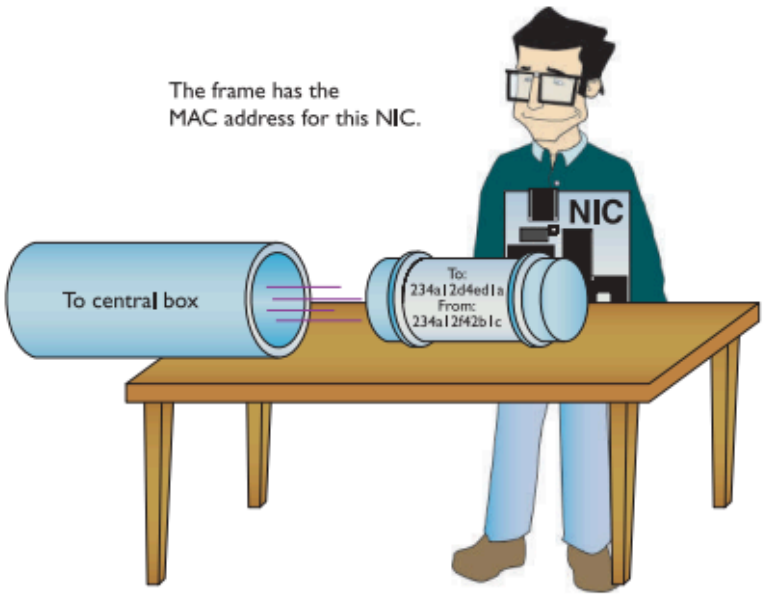


Figure 1-24
Reading an incoming frame

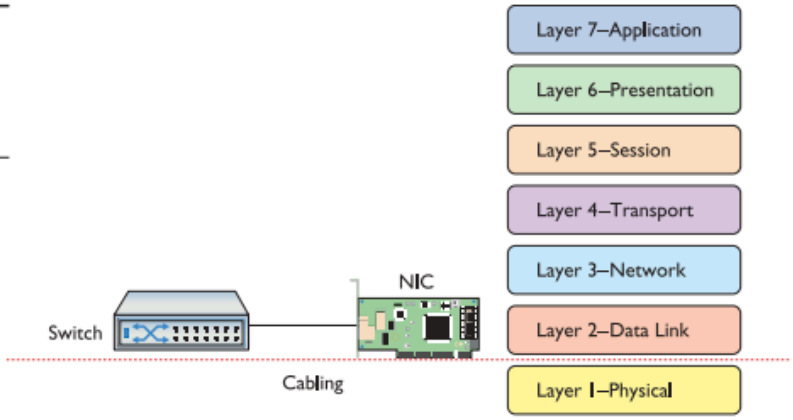


Figure 1-25
Layer 1 and Layer 2 are now properly applied to the network.

Note that the cabling (and hubs) are in the Physical layer. Switches handle traffic using MAC addresses, so they operate at Layer 2. That's the way modern wired networks work. The NIC is in the Data Link layer and the Physical layer.

The Two Aspects of NICs

Consider how data moves in and out of a NIC. On one end, frames move in and out of the NIC's network cable connection. On the other end, data moves back and forth between the NIC and the network operating system software. The many steps a NIC performs to keep this data moving—sending and receiving frames over the wire, creating outgoing frames, reading

incoming frames, and attaching MAC addresses—are classically broken down into two distinct jobs.

NOTE Sending NICs break frames into ones and zeroes for transmission; receiving NICs rebuild the frame on receipt. You get the idea.

The first job is called the *Link Logic Control (LLC)*. The LLC is the aspect of the NIC that talks to the system's operating system (usually via device drivers). The LLC handles multiple network protocols and provides flow control.

EXAM TIP The CompTIA Network+ exam tests you on the details of the OSI seven-layer model, so remember that the Data Link layer is the only layer that has sublayers.

The second job is called the *Media Access Control (MAC)*, which creates and addresses the frame. It adds the NIC's own MAC address and attaches MAC addresses to the frames. Recall that each frame the NIC creates must include both the sender's and recipient's MAC addresses. The MAC sublayer adds or checks the FCS. The MAC also ensures that the frames, now complete with their MAC addresses, are then sent along the network cabling. Figure 1-26 shows the Data Link layer in detail.

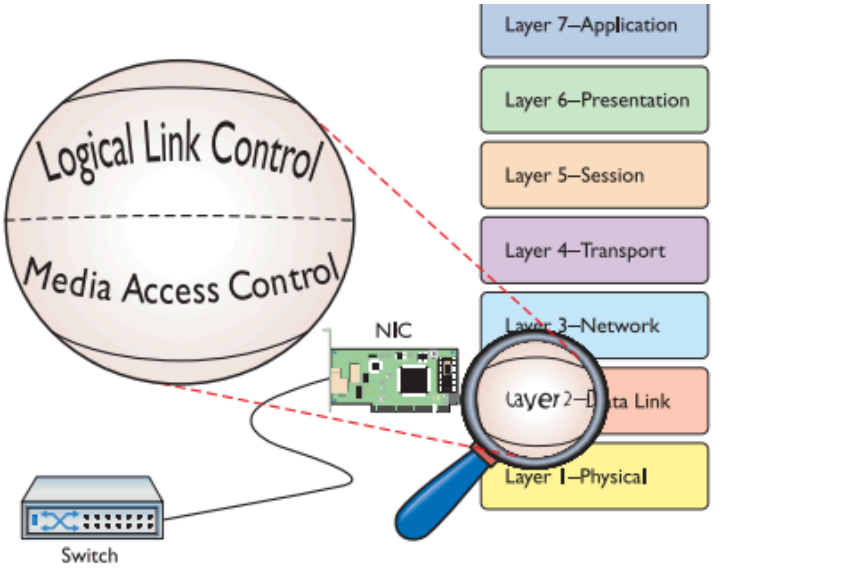


Figure 1-26 LLC and MAC, the two parts of the Data Link layer

NIC and Layers

Most networking materials that describe the OSI seven-layer model put NICs squarely into the Data Link layer of the model. It's at the MAC sublayer, after all, that data gets encapsulated into a frame, destination and source MAC addresses get added to that frame, and error checking occurs. What bothers most students with placing NICs solely in the Data Link layer is the

obvious other duty of the NIC—putting the ones and zeroes on the network cable for wired networks and in the air for wireless networks. How much more physical can you get? Many teachers will finesse this issue by defining the Physical layer in its logical sense— that it defines the rules for the ones and zeroes—and then ignore the fact that the data sent on the cable has to come from something. The first question when you hear a statement like that—at least to me—is, “What component does the sending?” It’s the NIC, of course, the only device capable of sending and receiving the physical signal. NICs, therefore, operate at both Layer 2 and Layer 1 of the OSI seven-layer model.

Beyond the Single Wire - Network Software and Layers 3-7

Getting data from one system to another in a simple network (defined as one in which all the computers connect to one switch) takes relatively little effort on the part of the NICs. But one problem with simple networks is that computers need to broadcast to get MAC addresses. It works for small networks, but what happens when the network gets big, like the size of the entire Internet? Can you imagine millions of computers all broadcasting? No data could get through. Equally important, data flows over the Internet using many technologies, not just Ethernet. These technologies don’t know what to do with Ethernet MAC addresses. When networks get large, you can’t use the MAC addresses anymore. Large networks need a *logical addressing* method, like a postal code or telephone numbering scheme, that ignores the hardware and enables you to break up the entire large network into smaller networks called *subnets*. Figure 1-27 shows two ways to set up a network. On the left, all the computers connect to a single switch. On the right, however, the LAN is separated into two five-computer subnets. To move past the physical MAC addresses and start using logical addressing requires some special software called a *network protocol*. Network protocols exist in every operating system. A network protocol not only has to create unique identifiers for each system, but also must create a set of communication rules for issues like how to handle data chopped up into multiple packets and how to ensure those packets get from one subnet to another. Let’s take a moment to learn a bit about the most famous collection of network protocols—TCP/IP—and its unique universal addressing system.

EXAM TIP MAC addresses are also known as *physical addresses*.

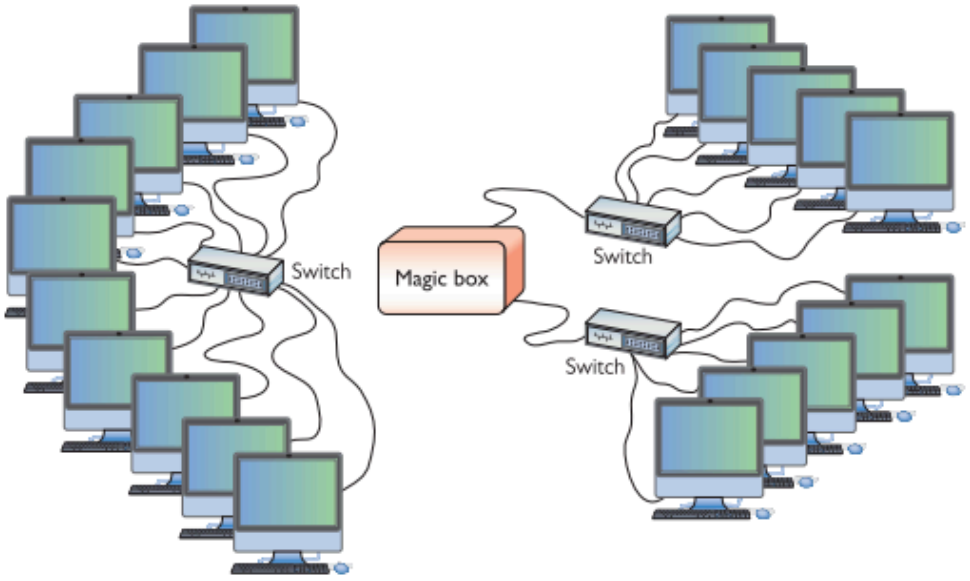


Figure 1-27 Large LAN complete (left) and broken up into two subnets (right)

Figure 1-27: Large LAN complete (left) and broken up into two subnets (right).

To be accurate, TCP/IP is really several network protocols designed to work together— better known as a *protocol suite* – but two protocols, TCP and IP, do so much work that the folks who invented all these protocols named the whole thing TCP/IP. *TCP* stands for *Transmission Control Protocol*, and *IP* stands for *Internet Protocol.* IP is the network protocol I need to discuss first; rest assured, however, I'll cover TCP in plenty of detail later.

IP - Playing on Layer 3, the Network Layer

At the *Network layer*, Layer 3, containers called *packets* get created and addressed so they can go from one network to another. The Internet Protocol is the primary logical addressing protocol for TCP/IP. IP makes sure that a piece of data gets to where it needs to go on the network. It does this by giving each device on the network a unique numeric called an *IP address.* An IP address is known as a *logical address* to distinguish it from the physical address, the MAC address of the NIC.

NOTE: A packet is the PDU for Layer 3.

IP uses a rather unique dotted decimal notation based on four 8-bit numbers. Each 8-bit number ranges from 0 to 255, and the four numbers are separated by three periods. (If you don't see how 8-bit numbers can range from 0 to 255, don't worry—by the end of this book, you'll understand these numbering conventions in more detail than you ever believed possible!) A typical IP address might look like this: 192.168.4.232

NOTE TCP/IP dominates networking today, and although it might be fun to imagine that it had humble beginnings in someone's garage lab, that's not the case. In the early 1970s, two researchers at the U.S. Defense Advanced Research Projects Agency (DARPA), Robert E. Kahn and Vinton Cerf, worked out the basic parameters of what would become TCP/IP. TCP/IP offered amazing robustness in its design and eventual implementation. Government research at its most profound and world shaping!

No two systems on the same network share the same IP address; if two machines accidentally receive the same address, unintended side effects may occur. These IP addresses don't just magically appear—they must be configured by the network administrator. What makes logical addressing powerful is another magic box—called a router— that connects each of the subnets, as previously shown in Figure 1-27. Routers use the IP address, not the MAC address, to forward data. This enables networks to connect across data lines that don't use Ethernet, like the telephone network. Each network type (such as Ethernet, SONET, and others that we'll discuss later in the book) uses a unique frame. Figure 1-28 shows a typical router.

Figure 1-28
Typical small router

In a TCP/IP network, each system has two unique identifiers: the MAC address and the IP address. The MAC address (the physical address) is literally burned into the chips on the NIC, whereas the IP address (the logical address) is simply stored in the system's software. MAC addresses come with the NIC, so you don't configure MAC addresses, whereas you must configure IP addresses using software. Figure 1-29 shows the MHTechEd network diagram again, this time with the MAC and IP addresses displayed for each system.

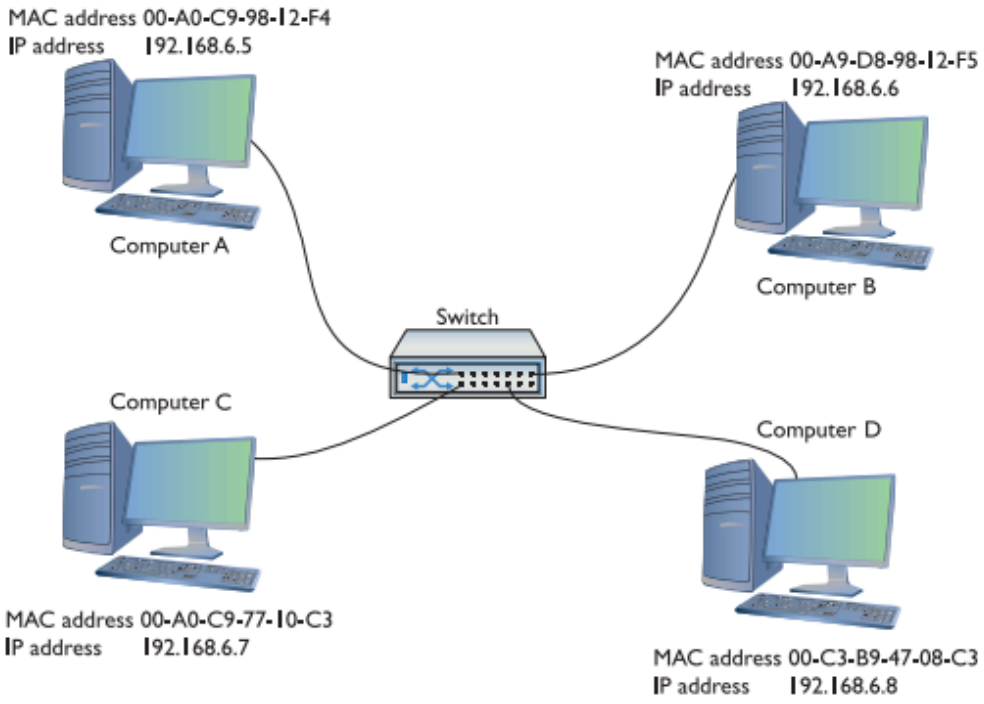


Figure 1-29 MHTechEd addressing

Figure 1-29 MHTechEd addressing

NOTE Try to avoid using redundant expressions. Even though many techs will say "IP protocol," for example, you know that "IP" stands for "Internet Protocol." It wouldn't be right to say "Internet Protocol protocol" in English, so it doesn't work in network speak either. (Also, don't say "NIC card" for the same reason!)

Packets within Frames For a TCP/IP network to send data successfully, the data must be wrapped up in two distinct containers. A frame of some type enables the data to move from one device to another. Inside that frame are both an IP-specific container that enables routers to determine where to send data—regardless of the physical connection type—and the data itself. In TCP/IP, that inner container is the packet. Figure 1-30 shows a typical IP packet; notice the similarity to the frames you saw earlier.

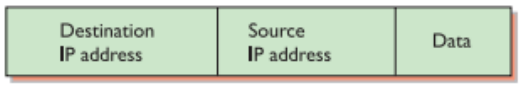


Figure 1-30 IP packet

Figure 1-30 IP Packet

NOTE This is a highly simplified IP Packet. I am not including lots of little parts of the IP packet in this diagram because they are not important to what you need to understand right now—but don't worry, you'll see them later in the book!

But IP packets don't leave their PC home without any clothes on! Each IP packet is handed to the NIC, which then encloses the IP packet in a regular frame, creating, in essence, a *packet within a frame.* I like to visualize the packet as an envelope, with the envelope in the pneumatic canister frame, as depicted in Figure 1-31. A more conventional drawing would look like Figure 1-32.

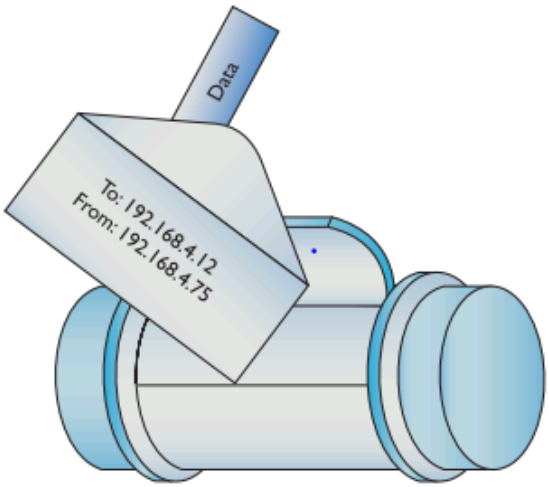


Figure 1-31 IP packet in a frame (as a canister)

Figure 1-31 IP packet in a frame (as a canister)

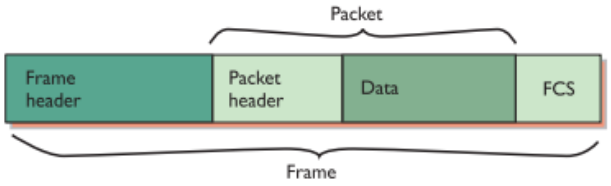


Figure 1-32 IP packet in a frame

Figure 1-32 IP packet in a frame

When you send data from one computer on a TCP/IP network such as the Internet, that data can go through multiple routers before it reaches its destination. Each router strips off the

incoming frame, determines where to send the data according to the IP address in the packet, creates a new frame, and then sends the packet within a frame on its merry way. The new frame type will be the appropriate technology for whatever connection technology connects to the next router. That could be a cable or DSL network connection, for example (Figure 1-33). The IP packet, on the other hand, remains unchanged.

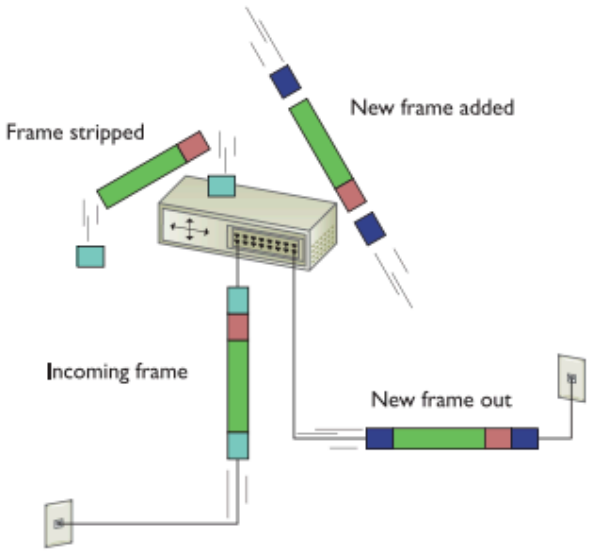


Figure 1-33 Router removing network frame and adding one for the outgoing connection

Figure 1-33 Router removing network frame and adding one for the outgoing connection

Once the packet reaches the destination subnet's router, that router strips off the incoming frame – no matter what type – looks at the destination IP address, and then adds a frame with the appropriate destination MAC address that matches the destination IP address. The receiving NIC strips away the Ethernet frame and passes the remaining packet off to the software. The networking software built into your operating system handles all the rest of the work. The NIC's driver software is the interconnection between the hardware and the software. The NIC driver knows how to communicate with the NIC to send and receive frames, but it can't do anything with the packet. Instead, the NIC driver hands the packet off to other services that know how to deal with all the separate packets and turn them into Web pages, e-mail messages, files, and so forth.

Segmentation and Assembly - Layer 4, The Transport Layer

Because most chunks of data are much larger than a single packet, they must be chopped up before they can be sent across a network. When a serving computer receives a request for some data, it must be able to chop the requested data into chunks that will fit into a packet (and eventually into the NIC's frame), organize the packets for the benefit of the receiving system, and hand them to the NIC for sending. This is called *segmentation*. The receiving system does the *reassembly* of the packets. It must recognize a series of incoming packets as one data

transmission, reassemble the packets correctly based on information included in the packets by the sending system, and verify that all the packets for that piece of data arrived in good shape. This part is relatively simple—the transport protocol breaks up the data into chunks called segments and gives each segment some type of sequence number. (*Datagrams* – also created at Layer 4—are simpler and don't get broken up into chunks or get sequence numbers.) I like to compare this sequencing process to the one that my favorite international shipping company uses. I receive boxes from UPS almost every day; in fact, some days I receive many, many boxes from UPS. To make sure I get all the boxes for one shipment, UPS puts a numbering system, like the one shown in Figure 1-34, on the label of each box. A computer sending data on a network does the same thing. Embedded into the data of each packet containing a segment is a sequencing number. By reading the sequencing numbers, the receiving system knows both the total number of segments and how to put them back together.

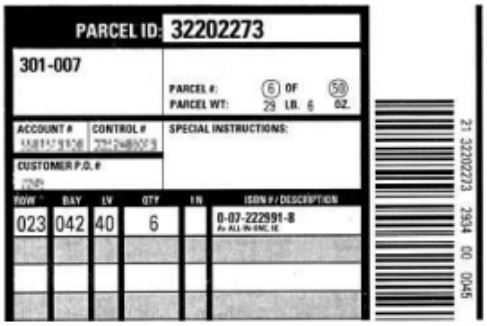


Figure 1-34 Labeling the boxes

Figure 1-34 Labelling the boxes

The MHTechEdNetwork keeps on getting more and more complex, doesn't it? And the Word document still hasn't been copied, has it? Don't worry, you're almost there – just a few more pieces to go!

Layer 4, the *Transport layer* of the OSI seven-layer model, has a big job: it's the segmentation/reassembly software. As part of its job, the Transport layer also initializes requests for packets that weren't received in good order (Figure 1-35).

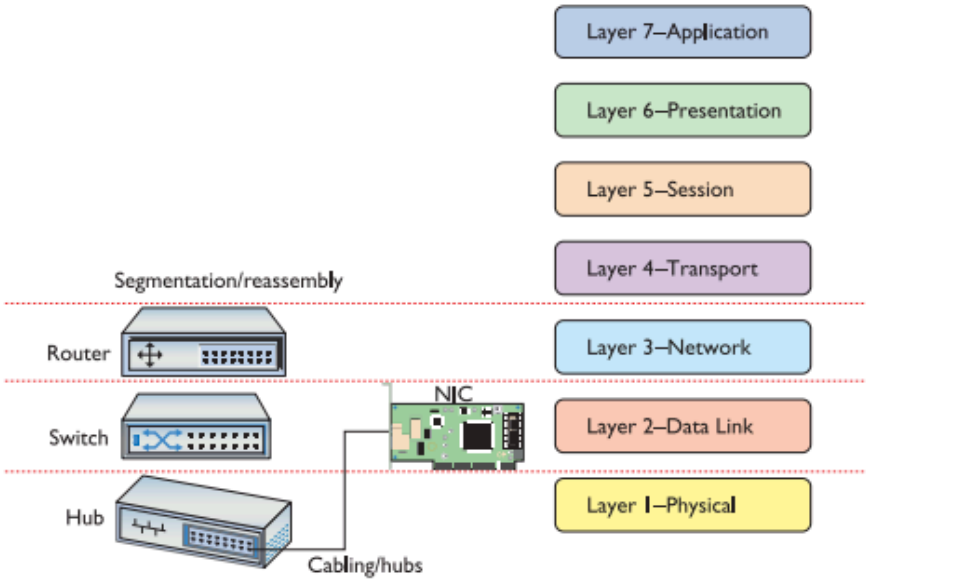


Figure 1-35 OSI updated

Figure 1-35 OSI updated

Connection-oriented vs Connectionless connection

Some protocols, like the Simple Mail Transfer Protocol (SMTP) used for sending e-mail messages, require that the e-mail client and server verify that they have a connection before a message is sent (Figure 1-36). This makes sense because you don't want your e-mail message to be a corrupted mess when it arrives.

Figure 1-36 Connection between e-mail client and server

Figure 1-36 Connection between e-mail client and server

Alternatively a number of TCP/IP protocols simply send data without waiting to verify that the receiving system is ready (Figure 1-37). When using voice over IP (VoIP), for example, the call is made without verifying first whether another device is there.

for example, the call is made without verifying first whether another device is there.

Figure 1-37 Connectionless communication

The connection-oriented protocol is *Transmission Control Protocol (TCP)*. The connectionless protocol is *User Datagram Protocol (UDP)*.

NOTE Chapter 6 covers TCP, UDP, and all sorts of other protocols in detail.

Everything you can do on the Internet, from Web browsing to Skype phone calls to playing World of Warcraft, is predetermined to be either connection-oriented or connectionless. It's simply a matter of knowing your applications.

Segments within Packets To see the Transport layer in action, strip away the IP addresses from an IP packet. What's left is a chunk of data in yet another container called a *TCP segment*. TCP segments have many other fields that ensure the data gets to its destination in good order. These fields have names such as Source port, Destination port, Sequence number, and Acknowledgment number. Figure 1-38 shows a typical (although simplified) TCP segment.

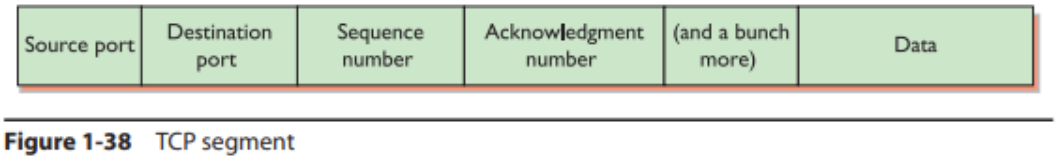


Figure 1-38 TCP segment

Figure 1-38 TCP segment

Chapter 6 goes into more detail on TCP segments, but let's look at source and destination ports as an example. You saw physical ports earlier in the chapter, but this use of the word "port" means something completely different. In this context, a *port* – a number between 1 and 65,536—is a logical value assigned to specific applications or services. A quick example will make this clear. Many TCP segments come into any computer. The computer needs some way to determine which TCP segments go to which applications. A Web server, for example, sees a lot of traffic, but it "listens" or looks for TCP segments with the destination port numbers 80 or 443, grabs those segments, and processes them. Equally, every TCP segment contains another port number—the source port—so the client knows what to do with returning information. Data comes from the Application layer (with perhaps some input from Presentation and Session). The Transport layer breaks that data into chunks, adding port numbers and sequence numbers, creating the TCP segment. The Transport layer then hands the TCP segment to the Network layer, which, in turn, creates the IP packet. Although a lot of traffic on a TCP/IP network uses TCP at the Transport layer, like Yoda said in *The Empire Strikes Back*, "There is another," and that's UDP. Following the same process, the Transport layer adds port and length numbers plus a checksum as a header and combines with data to create a container called a *UDP datagram*. A UDP datagram lacks most of the extra fields found in TCP segments, simply because UDP doesn't care if the receiving computer gets its data. Figure 1-39 shows a UDP datagram.

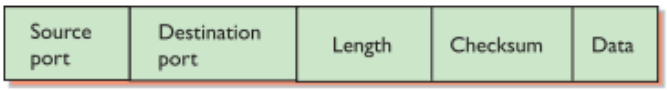


Figure 1-39 UDP datagram

Figure 1-39 UDP datagram

Talking on a Network – Layer 5, the Session Layer

Now that you understand that the system uses software to segment and reassemble data packets, what's next? In a network, any one system may be talking to many other systems at any given moment. For example, Shannon's PC has a printer used by all the MHTechEd systems, so there's a better than average chance that, as Scott tries to access the Word document, another system will be sending a print job to Shannon's PC (Figure 1-40). Shannon's system must direct these incoming files, print jobs, Web pages, and so on, to the right programs (Figure 1-41). Additionally, the operating system must enable one system to make a connection to another system to verify that the other system can handle whatever operation the initiating system wants to perform. If Bill's system wants to send a print job to Shannon's printer, it first contacts Shannon's system to ensure that it is ready to handle the print job. The *session software* handles this part of networking, connecting applications to applications.

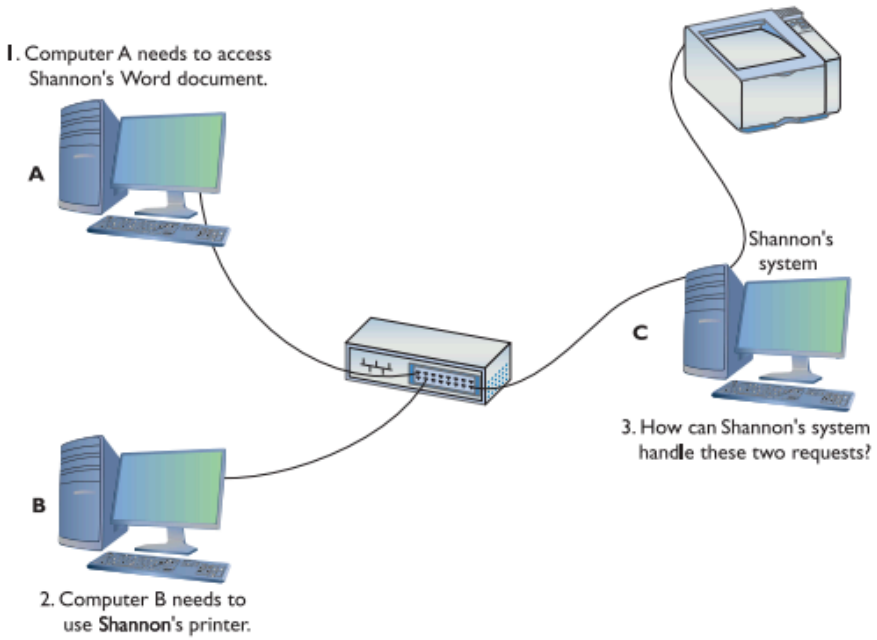


Figure 1-40 Handling multiple inputs

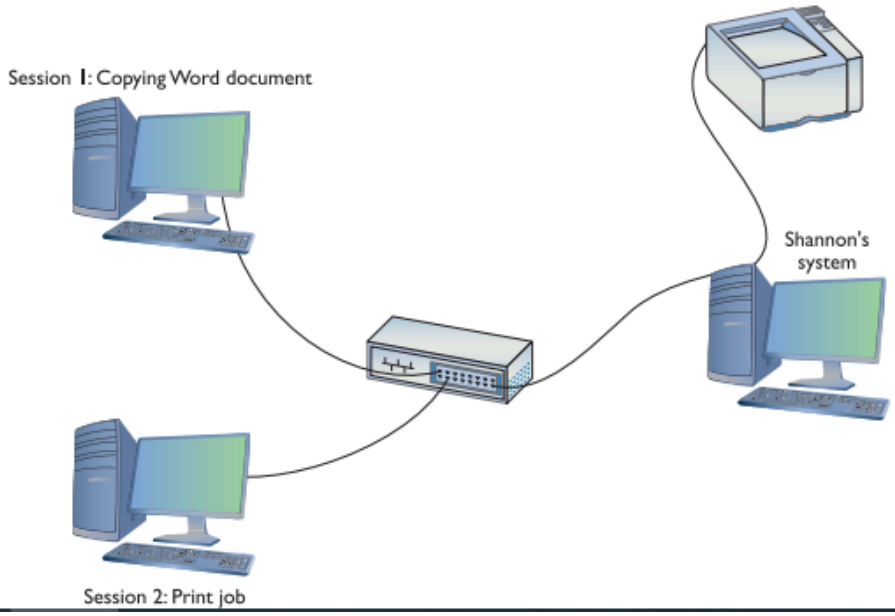


Figure 1-41 Each request becomes a session.

How many sessions does a typical system have running at one time? Well, if you have a TCP/IP network (and who doesn't these days), you can run the netstat program from a command prompt to see all of them. Open a command prompt and type the following:

Netstat -a

Then press the ENTER key to see your sessions. Don't worry about trying to interpret what you see—Chapter 8 covers netstat in detail. For now, simply appreciate that each line in the netstat output is a session. Count them! (You can also try the ss command in Linux to view sessions.) Layer 5, the *Session Layer* of the OSI seven-layer model, handles all the sessions for a system (Figure 1-42). The Session layer initiates sessions, accepts incoming sessions, and opens and closes existing sessions.

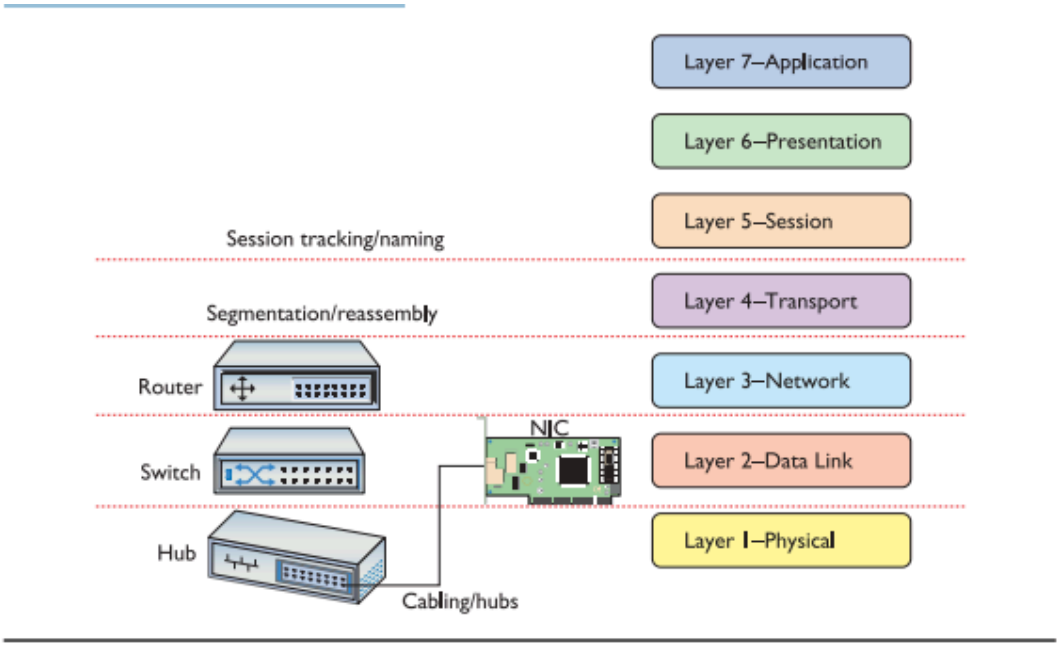


Figure 1-42 OSI updated

Figure 1-42 OSI updated

Many operating systems represent a session using the combination of the IP address and port numbers for both sides of a TCP or UDP communication. You can see a Web browser's session connecting to a Web server, for example, by running *netstat -n.*
It'll return many lines like this:

TCP 192.168.4.34:45543 11.12.13.123:80 Established

The numbers describe the session. A Web client with IP address 192.168.4.34, using port number 45543, is in a TCP session with a Web server (we know it's a Web server because port 80 is dedicated to Web servers) using IP address 11.12.13.123.

Translation – Layer 6, the Presentation Layer

The *Presentation Layer* translates data from lower layers into a format usable by the Application layer, and vice versa (Figure 1-43). This manifests in several ways and isn't

necessarily clear-cut. The messiness comes into play because TCP/IP networks don't necessarily map directly to the OSI model.

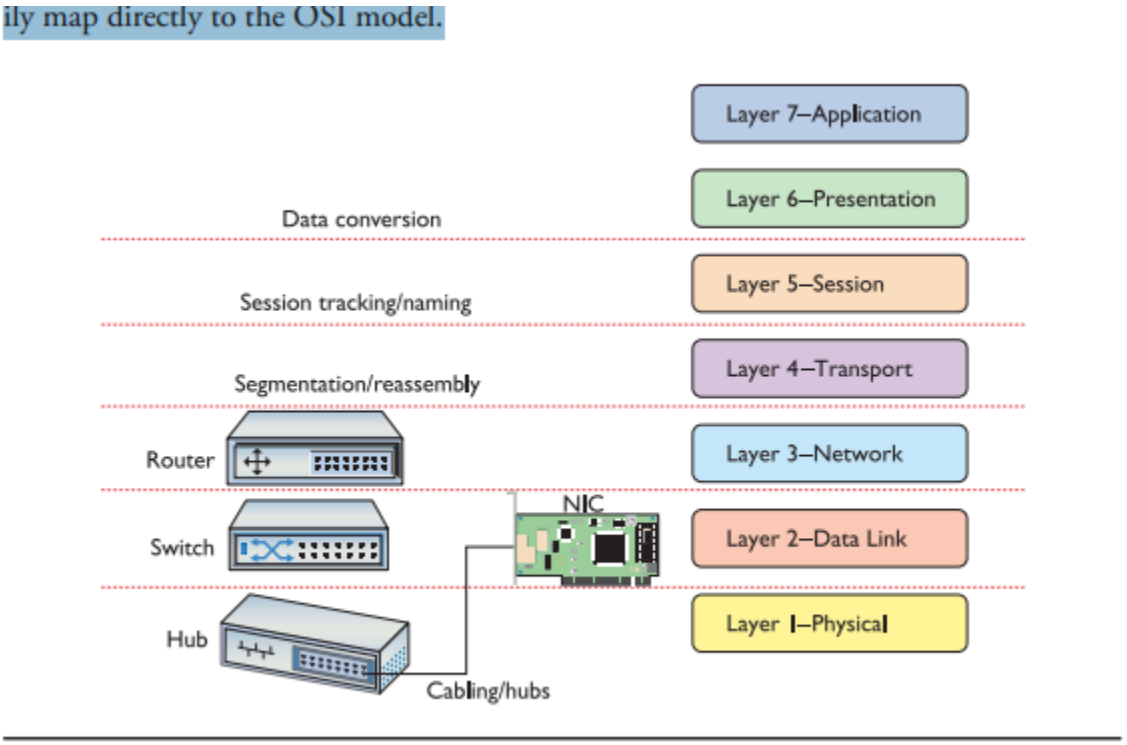


Figure 1-43 OSI updated

A number of protocols function on more than one OSI layer and can include Layer 6, Presentation. The encryption protocol used in e-commerce, Transport Layer Security (TLS), for example, seems to initiate at Layer 5 and then encrypt and decrypt at Layer 6. But even one of the authors of the protocol disputes that it should even be included in any OSI chart! It makes for some confusion.

Network Applications - Layer 7, the Application Layer

The last and most visible part of any network is the software applications that use it. If you want to copy a file residing on another system in your network, you need an applet like Network in Windows 10 that enables you to access files on remote systems. If you want to view Web pages, you need a Web browser like Google Chrome or Mozilla Firefox. The people who use a network experience it through an application. A user who knows nothing about all the other parts of a network may still know how to open an e-mail application to retrieve mail (Figure 1-44). Applications may include additional functions, such as encryption, user authentication, and tools to control the look of the data. But these functions are specific to the given applications. In other words, if you want to put a password on your Word document, you must use the password functions in Word to do so.

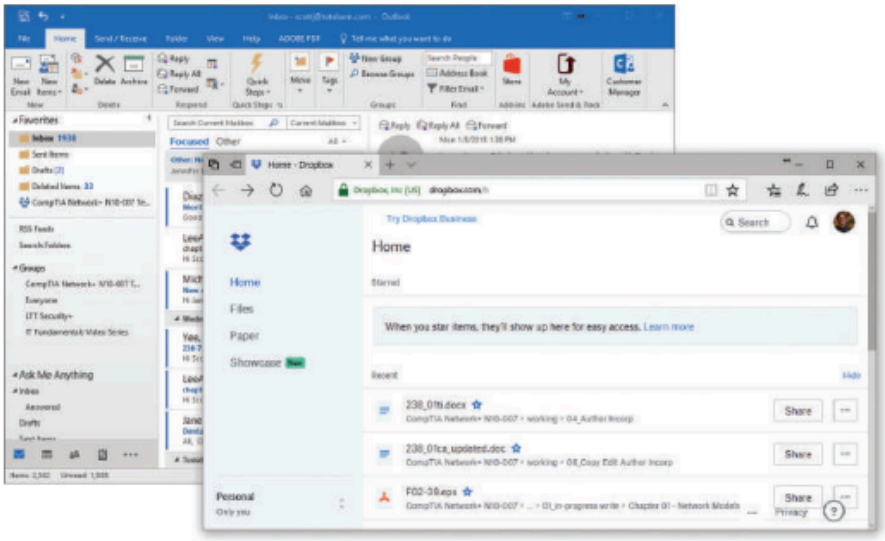


Figure 1-44 Network applications at work

Figure 1-44 Network applications at work

The *Application layer* is Layer 7 in the OSI seven-layer model. Keep in mind that the Application layer doesn't refer to the applications themselves. It refers to the code built into all operating systems that enables network-aware applications. All operating systems have *Application Programming Interfaces (APIs)* that programmers can use to make their programs network aware (Figure 1-45). An API, in general, provides a standard way for programmers to enhance or extend an application's capabilities.

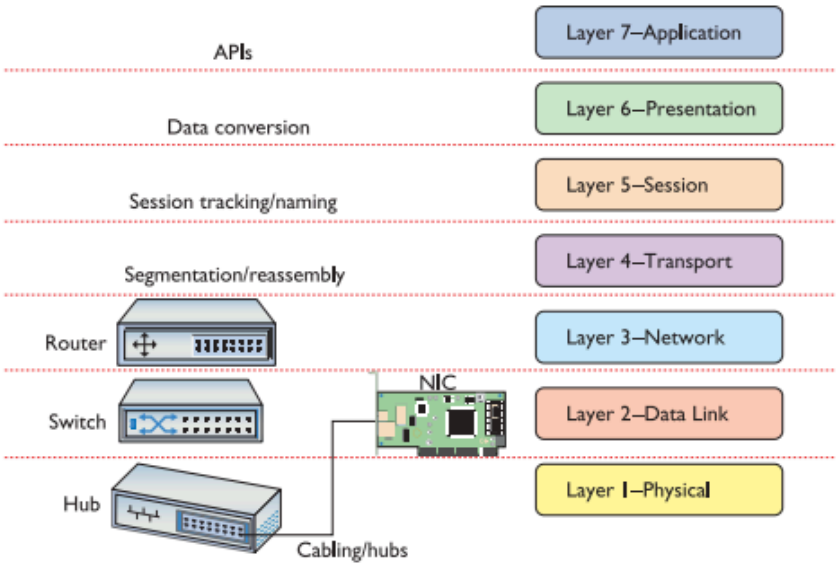


Figure 1-45 OSI updated

Figure 1-45 OSI updated

Encapsulation and Decapsulation

The term *encapsulation* encompasses the entire process of preparing data to go onto a network. This includes all the steps from the application to the Application, Presentation, Session, Transport, Network, and Data Link layers. Each layer adds more information so that the data gets to the correct recipient and the recipient knows what to do with the data. The receiving computer reverses the process, stripping all the extra header information out as the data goes up the stack. This reverse process is called decapsulation. The Transport layer creates a segment or datagram and hands it down to the Network layer. That layer adds IP information, encapsulating the segment or datagram. The Data Link layer wraps up all that goodness, encapsulating the packet in a frame for delivery over the network. The NIC turns the frame into bits and bytes for transmission over the network medium.

The OSI Seven-Layer Model and Remote Work

Beth works remotely for a large company as a data analyst, putting her advanced degree in information science to good use. Let's explore her typical workflow in this section to see how the OSI seven-layer model applies. Beth connects to the Internet wirelessly with her laptop. Her company, like so many these days, uses a number of different online services to help coordinate its far-flung workforce. These services go by names like Microsoft 365, Dropbox, GitHub, and so forth, but fundamentally these services all live on the Web and so Beth spends much of her workday in her browser of choice, Firefox (Figure 1-46). In this scenario, Beth's computer isn't plugged into an Ethernet port. There are no local wires. Her laptop doesn't even have an Ethernet port (because it's modern and skinny). Do OSI layers even apply? If you answered, "Of

course," then you win a prize because they most definitely apply. The wireless radio waves that connect Beth's laptop to a wireless access point (WAP) operate at Layer 1 (Figure 1-47). So too do all the physical wires connecting the WAP to her router, Internet service provider (ISP), and all the routers in between there and her corporate network and other Internet-based services.

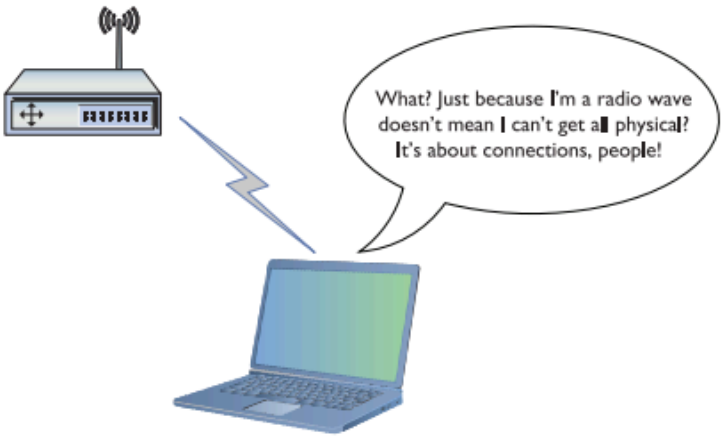


Figure 1-47 Wireless is "physical" too!

The wireless NIC in Beth's laptop most certainly has a MAC address and connects to the WAP with frames; the WAP uses MAC addresses to connect to the local switch. That's all clearly Layer 2 happiness. Beth conducts almost all of her work via the Web and thus relies almost exclusively on TCP/IP for connections and interactions. By definition, her laptop must have a valid IP address or two (Layer 3) and must encapsulate/decapsulate segments and datagrams at the Transport layer (Layer 4). But the heavy lifting happens at Layer 7 with HTTP and TLS (HTTPS), because Web-based tools today rely on those protocols (as well as others). Figure 1-48 shows that the tools Beth uses all work with Layer 7. The bottom line is that the OSI seven-layer model provides you with a way to conceptualize a network to determine what could cause a specific problem when the inevitable problems occur. Good techs always use a model to troubleshoot their networks. The OSI model can apply to a simple network or a more advanced network. If Beth can't print to a networked printer, for example, a model can help solve the problem. If her NIC shows activity, then, using the OSI model, you can set aside both the Physical layer (Layer 1) and Data Link layer (Layer 2). You'll find yourself moving up the layer ladder to the OSI model's Network layer (Layer 3). If her computer has a proper IP address, then you can set that layer aside too, and you can move on up to check other layers to solve the problem.

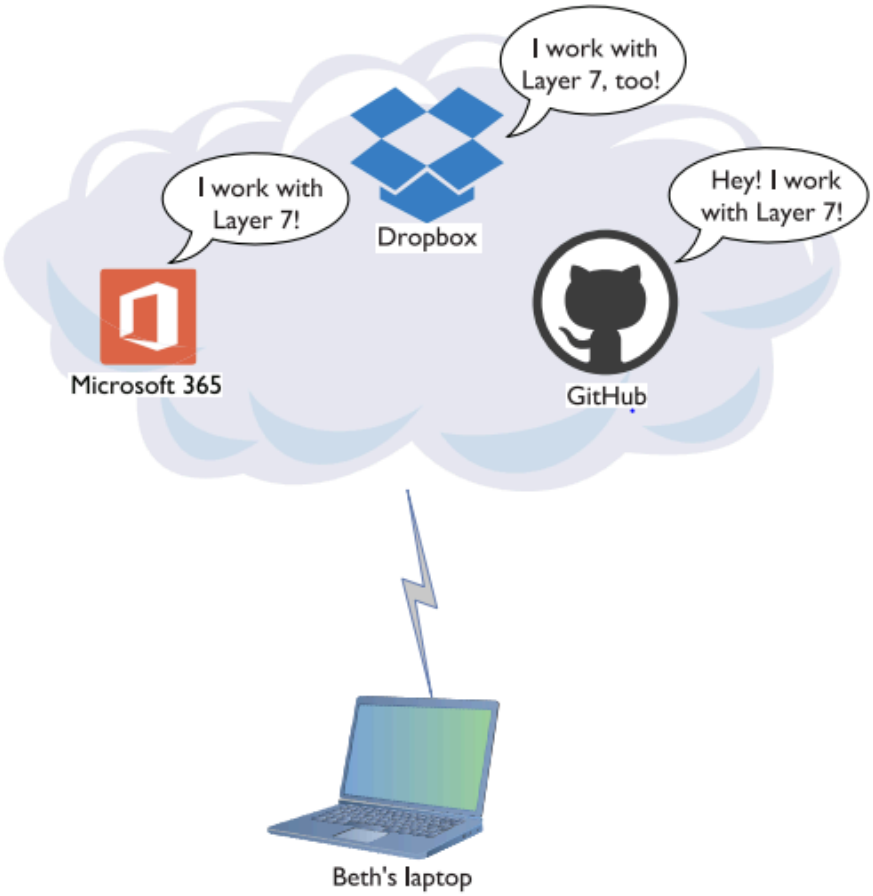


Figure 1-48 Beth's productivity tools

Understanding the OSI model is important. It is the primary diagnostic tool for troubleshooting networks and also the communication tool for talking with your fellow techs.

EXAM TIP Beth accesses servers to do her job; her laptop is a client of those servers. Thus, the previous scenario describes a classic *client-server* network type. In some circumstances, however, Beth might access resources distributed on many computers. In turn, her computer might share some of those resources with others. This alternative network type, typified by the BitTorrent file sharing protocol, is called *peer-to-peer*. Look for a question on the CompTIA Network+ exam that contrasts client-server and peer-to-peer networking.

Chapter Review

Questions

1. Where does (did) a hub send data?
    a. Only to the receiving system
    b. Only to the sending system
    c. To all the systems connected to the hub
    d. Only to the server

2. What uniquely identifies every NIC?
    a. IP address
    b. Media access control address
    c. ISO number
    d. Media packet number
3. What uniquely identifies every NIC?
    a. IP address
    b. Media access control address
    c. ISO number
    d. Packet ID number
4. Which Windows utility do you use to find the MAC address for a system?
    a. Ipconfig /all
    b. Ipcfg /all
    c. ping
    d. mac
5. A MAC address is also known as a(n) _____ address.
    a. IP
    b. Logical
    c. Physical
    d. OEM
6. A NIC sends data in discrete chunks called _____.
    a. segments
    b. sections
    c. frames
    d. Layers
7. The MAC address of which of the following begins a frame?
    a. Receiving system
    b. Sending system
    c. Network
    d. Router
8. A frame ends with a special bit called the frame check sequence (FCS). What does the FCS do?
    a. Cycles across the network
    b. Verifies that the MAC addresses are correct
    c. Verifies that the data arrived correctly
    d. Verifies that the IP address is correct
9. Which layer of the OSI model controls the segmentation and reassembly of data?
    a. Application layer
    b. Presentation layer
    c. Session layer
    d. Transport layer
10. Which layer of the OSI model keeps track of a system's connections to send the right response to the right computer?
    a. Application layer

b. Presentation layer
c. Session layer
d. Transport layer

Chapter 2 - Cabling and Topology

The CompTIA Network+ certification exam expects you to know how to

- 1.2 Explain the characteristics of network topologies and network types
- 1.3 Summarize the types of cables and connectors and explain which is the appropriate type for a solution
- 5.2 Given a scenario, troubleshoot common cable connectivity issues and select the appropriate tools

To achieve these goals, you must be able to:

- Explain the different types of network topologies
- Describe the different types of network cabling and connectors
- Describe the IEEE networking standards

Every network must provide some method to get data from one system to another. In most cases, this method consists of some type of cabling running between systems, although most networks today incorporate wireless methods to move data as well. Stringing those cables brings up a number of critical issues you need to understand to work on a network. How do all these cables connect the computers? Does every computer on the network run a cable to a central point? Does a single cable snake through the ceiling, with all the computers on the network connected to it? These questions need answering! Furthermore, manufacturers need standards so they can make networking equipment that works well together. While we're talking about standards, what about the cabling itself? What type of cable? What quality of copper or fiber? How thick should it be? Who defines the standards for cables so they all work in the network? This chapter answers these questions in three parts. First, you will learn about the *network topology* – the way that pieces of hardware connect to one another, via wires or wirelessly. Second, you will tour the most common standardized cable types used in networking. Third, you will learn about the IEEE committees that create network technology standards.

Test Specific

Network Topologies

Computer networks employ many different topologies, or ways of connecting computers and other devices like switches and printers together. This section looks at both the historical topologies—bus, ring, and star; all long dead—and the modern topologies— hybrid and mesh. In addition, we will look at what parameters are used to make up a network topology.

NOTE Wireless technologies employ topologies too, just not with wires. We'll cover the common wireless topologies—infrastructure and ad hoc— in Chapter 14.

Bus and Ring

The first generation of wired networks used one of two topologies, both shown in Figure 2-1. A *bus topology* network used a single cable (i.e. *the bus*) that connected all of the computers in a line. A *ring topology* network connected all computers on the network with a ring of cable.

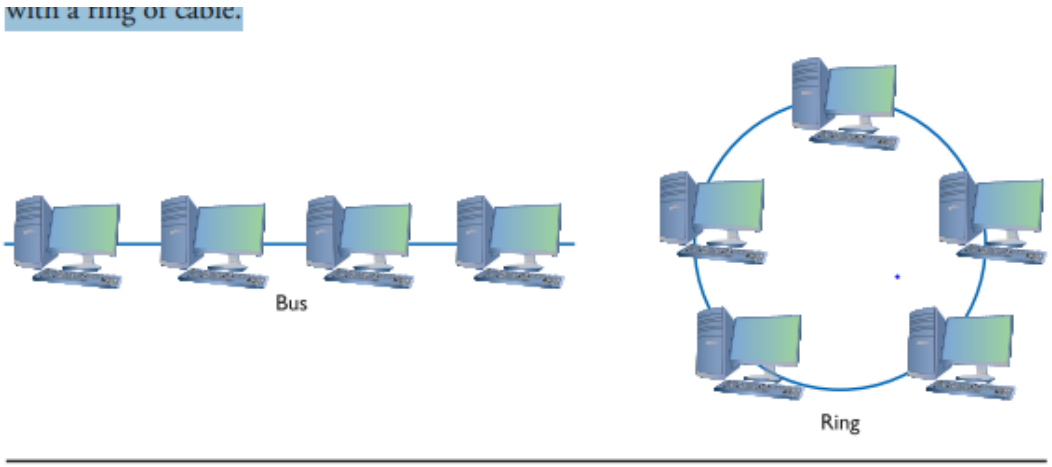


Figure 2-1 Bus and ring topologies

Figure 2-1 Bus and ring topologies

NOTE Topologies are diagrams, much like an electric circuit diagram. Real network cabling doesn't go in perfect straight circles or perfect straight lines.

Data flowed differently between bus and ring networks, creating different problems and solutions. In bus topology networks, data from each computer simply went out on the whole bus. A network using a bus topology needed termination at each end of the cable to prevent a signal sent from one computer from reflecting at the ends of the cable, quickly bringing the network down (Figure 2-2).

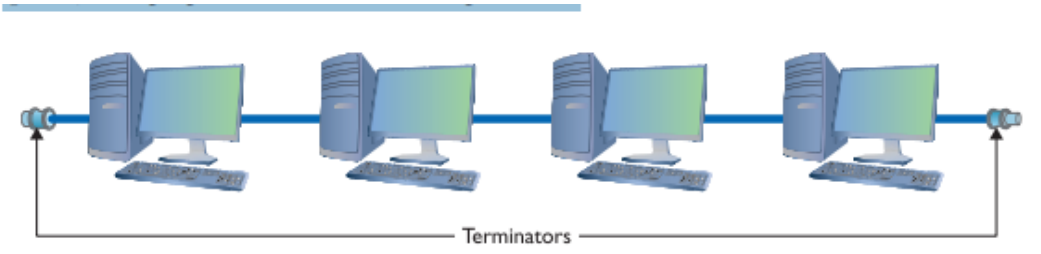


Figure 2-2 Terminated bus topology

Figure 2-2 Terminated bus topology

In a ring topology network, in contrast, data traffic moved in a circle from one computer to the next in the same direction (Figure 2-3). With no end to the cable, ring networks required no termination.

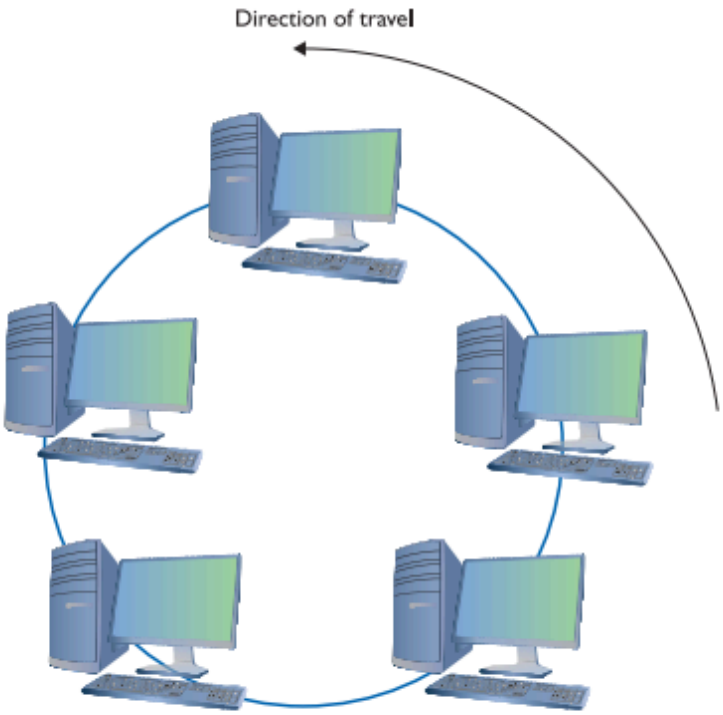


Figure 2-3
Ring topology moving in a certain direction

Bus and ring topology networks worked well but suffered from the same problem: the entire network stopped working if the cable broke at any point (Figure 2-4). The broken ends on a bus topology network didn't have the required termination, which caused reflection between computers that were still connected. A break in a ring topology network simply broke the circuit, stopping the data flow.

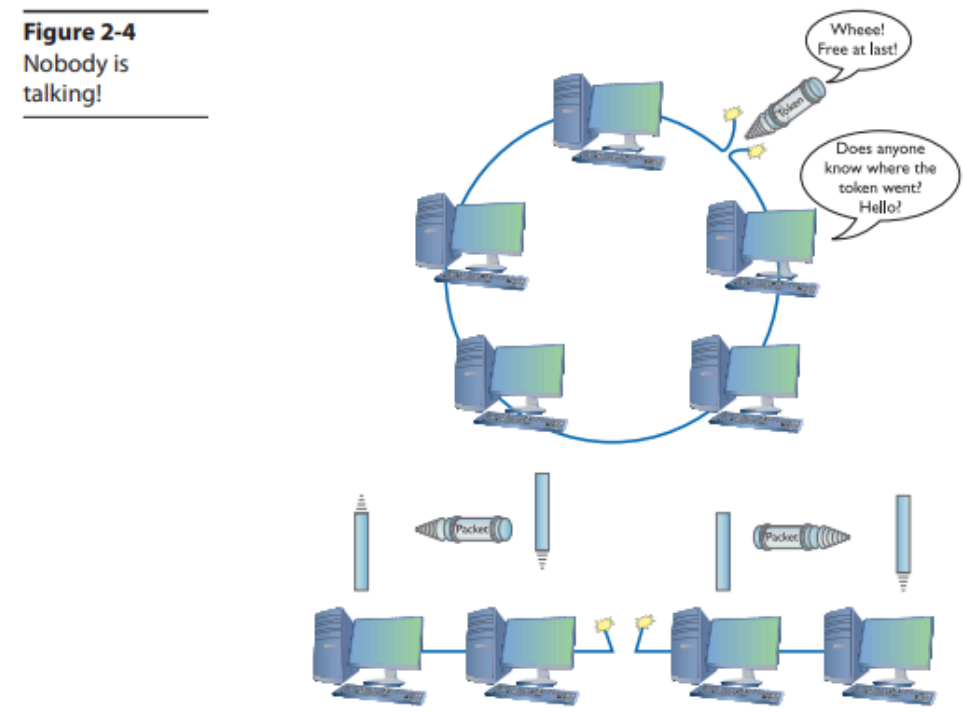


Figure 2-4 Nobody is talking!

Star

The *star topology*, also called *hub-and-spoke*, used a central connection box for all the computers on the network (Figure 2-5). Star topologies had a huge benefit over ring and bus topologies by offering *fault tolerance* – if one of the cables broke, all of the other computers could still communicate. Bus and ring topology networks were popular and inexpensive to implement, however, so the old-style star topology networks weren't very successful. Network hardware designers couldn't easily redesign their existing networks to use a star topology.

Figure 2-5
Star topology

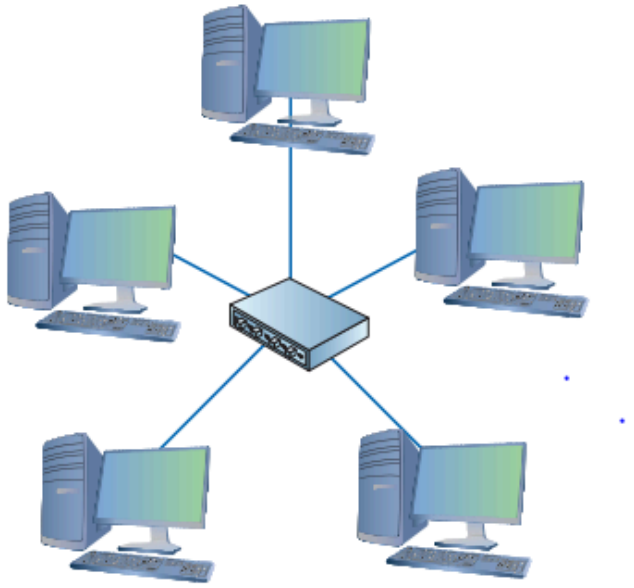


Figure 2-5 Star topology

Hybrid

Even though network designers couldn't easily use a star topology, the benefits of star topologies were overwhelming, motivating smart people to come up with a way to use star topologies without requiring a major redesign—and the way they did so was ingenious. The ring topology network designers struck first by taking the entire ring and shrinking it into a small box, as shown in Figure 2-6.

Figure 2-6
Shrinking the ring

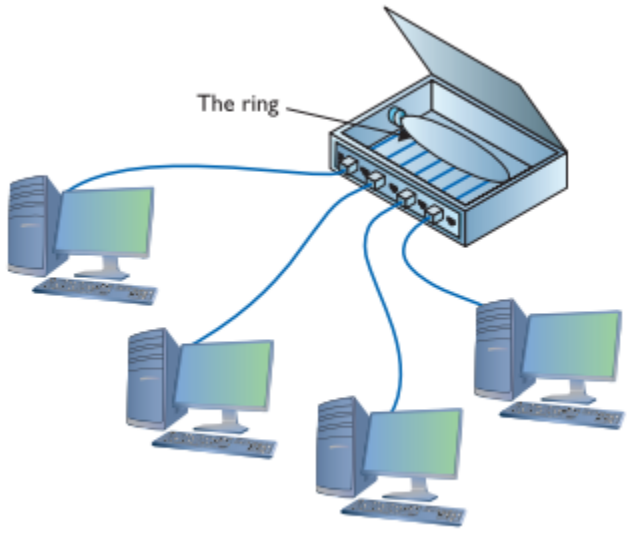
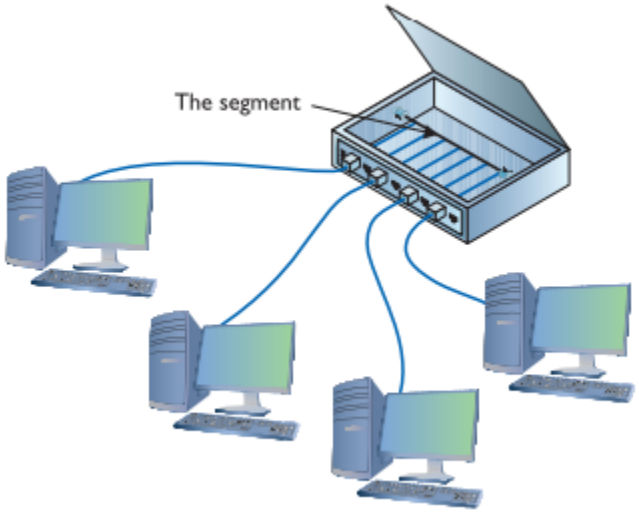


Figure 2-6 Shrinking the ring

This was quickly followed by the bus topology folks, who, in turn, shrunk their bus (better known as the *segment*) into their own box (Figure 2-7).

Figure 2-7
Shrinking the segment



Physically, both of these hybrid designs looked like a star, but if you examined them as an electronic schematic, the signals acted like a ring or a bus. Clearly the old definition of topology needed a little clarification. When we talk about topology today, we separate how the cables physically look (the *physical topology*) from how the signals travel electronically (the signaling topology or *logical topology*).

EXAM TIP

Most techs refer to the signaling topology as the logical topology today. That's how you'll see it on the CompTIA Network+ exam as well. Look for a question on the exam that challenges you on logical versus physical topology.

Any form of networking technology that combines a physical topology with a signaling topology is called a *hybrid topology*. Hybrid topologies have come and gone since the earliest days of networking. Only two hybrid topologies, *star-ring topology* and *star-bus topology*, ever saw any amount of popularity. Eventually, star-ring lost market share, and star-bus reigns as the undisputed "star" (pun intended) of wired network topologies.

NOTE

The most successful of the star-ring topology networks was called Token Ring, manufactured by IBM.

Mesh

Topologies aren't just for wired networks. Wireless networks also need topologies to get data from one machine to another, but using radio waves instead of cables involves somewhat different topologies. Wireless devices can connect in a mesh topology network, where every computer connects to every other computer via two or more routes. Some of the routes between two computers may require traversing through another member of the mesh network. (See Chapter 14 for the scoop on wireless network types.) There are two types of meshed topologies: partially meshed and fully meshed (Figure 2-8). In a *partially meshed network,* at least two machines have redundant connections. Every machine doesn't have to connect to every other machine. In a *fully meshed topology* network, every computer connects directly to every other computer.

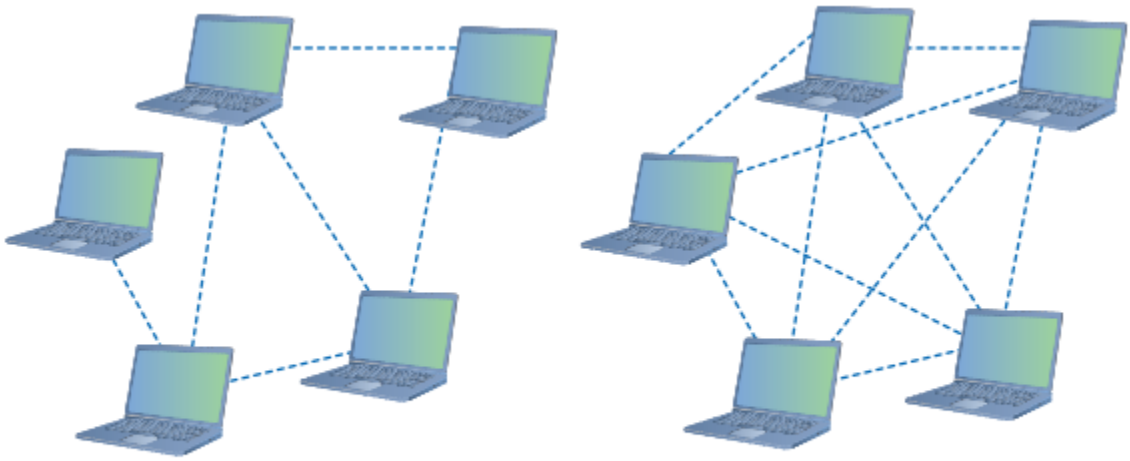


Figure 2-8 Partially and fully meshed topologies

Figure 2-8 Partially and fully meshed topologies

Parameters of a Topology

Although a topology describes the method by which systems in a network connect, the topology alone doesn't describe all of the features necessary to enable those networks. The term *bus topology*, for example, describes a network that consists of machines connected to the network via a single linear piece of cable. Notice that this definition leaves a lot of questions unanswered. What is the cable made of? How long can it be? How do the machines decide which machine should send data at a specific moment? A network based on a bus topology can answer these questions in several ways—but it's not the job of the topology to define issues like these. A functioning network needs a more detailed standard.

EXAM TIP Make sure you know the topologies: bus, ring, star/hub-and-spoke, hybrid, and mesh.

Over the years, manufacturers and standards bodies have created network technologies based on different topologies. A *network technology* is a practical application of a topology and other critical tools that provides a method to get data from one computer to another on a network. These network technologies have names like 100BASE-T, 1000BASE-LX, and 10GBASE-T. You will learn all about these in the next two chapters.

SIM Check out the excellent Chapter 2 "Topology Matching" Challenge! Over at http://totalsem.com/008. It's a good tool for reinforcing the topology variations.

Cabling and Connectors

Most networked systems link together using some type of cabling. Different types of networks over the years have used different types of cables—and you need to learn about all these cables to succeed on the CompTIA Network+ exam. This section explores scenarios where you would use common network cabling. All cables used in the networking industry can be categorized in two distinct groups: copper and fiber-optic. All styles of cables have distinct connector types that you need to know.

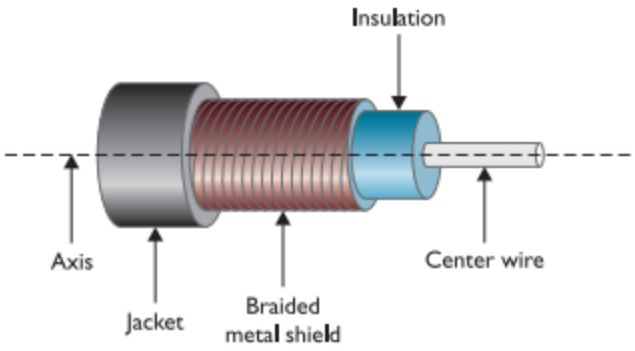
Copper Cabling and Connectors

The most common form of cabling uses copper wire wrapped up in some kind of protective sheathing, thus the term *copper cables*. The two primary types of copper cabling used in the industry are coaxial and twisted pair. Both cable types sport a variety of connector types. I'll cover the connector types as I discuss the cable varieties.

Coaxial Cable

Coaxial cable contains a central copper conductor wire surrounded by an insulating material, which, in turn, is surrounded by a braided metal shield. The cable is referred to as coaxial (coax for short) because the center wire and the braided metal shield share a common axis or centerline (Figure 2-9).

Figure 2-9
Cutaway view of coaxial cable



Coaxial cable shields data transmissions from interference. Many devices in the typical office environment—including lights, fans, copy machines, and refrigerators—generate magnetic fields. When a metal wire encounters these magnetic fields, electrical current is generated along the wire. This extra current, called *electromagnetic interference (EMI)*, shut down a network because it is easily misinterpreted as a signal by devices like NICs. To prevent EMI from affecting the network, the outer mesh layer of a coaxial cable shields the center wire (on which the data is transmitted) from interference (Figure 2-10).

Figure 2-10
Coaxial cable showing braided metal shielding

Figure 2-10 Coaxial cable showing braided metal shielding

Early bus topology networks used coaxial cable to connect computers together. Back in the day, the most popular cable used special bayonet-style connectors called *BNC connector s* (Figure 2-11).

Figure 2-11
BNC connector
on coaxial cable

Figure 2-11 BNC connector on coaxial cable

NOTE Techs all around the globe argue over the meaning of BNC. A solid percentage says with authority that it stands for “British Naval Connector.” An opposing percentage says with equal authority that it stands for “Bayonet Neill-Concelman,” after the stick-and-twist style of connecting and the purported inventors of the connector. The jury is still out, though this week I’m leaning toward Neill and Concelman and their bayonet-style connector.

You’ll find coaxial cable used today primarily to enable a cable modem to connect to an Internet service provider (ISP). That’s the typical scenario for using coaxial cable: connecting a computer to the cable modem enables that computer to access the Internet. This cable is the same type used to connect televisions to cable boxes or to satellite receivers. These cables use an *F-connector (*or *F-type connector*) that screws on, making for a secure connection (Figure 2-12).

Figure 2-12
F-type connector
on coaxial cable

EXAM TIP Coaxial cabling is also very popular for use with satellite dishes, over-the-air antennas, and even some home video devices. This book covers cable and other Internet connectivity options in great detail in Chapter 13.

Cable modems connect using one of two coaxial cable types. RG-59 was used primarily for cable television rather than networking. Its thinness and the introduction of digital cable motivated the move to the more robust *RG-6*, , the predominant cabling used today (Figure

2-13). All coax cables have a *Radio Guide (RG) rating*. The U.S. military developed these ratings to provide a quick reference for the different types of coax. The only important measure of coax cabling is its *Ohm* rating, a relative measure of the resistance (or more precisely, characteristic impedance) on the cable. You may run across other coax cables that don't have acceptable Ohm ratings, although they look just like network-rated coax. Both RG-6 and RG-59 cables are rated at 75 Ohms.

Figure 2-13
RG-6 cable

NOTE The Ohm rating of a piece of cable describes the impedance of that cable. *Impedance* describes a set of characteristics that define how much a cable resists the flow of electricity. This isn't simple resistance, though. Impedance is also a factor in such things as how long it takes the wire to get a full charge—the wire's *capacitance*– and more.

Given the popularity of cable for television and Internet in homes today, you'll run into situations where people need to take a single coaxial cable and split it. Coaxial handles this quite nicely with coaxial splitters like the one shown in Figure 2-14. You can also connect two coaxial cables together easily using a barrel connector when you need to add some distance to a connection (Figure 2-15). Table 2-1 summarizes the coaxial standards.

some distance to a connection (Figure 2-15). Table 2-1 summarizes the coaxial standards.

Figure 2-14
Coaxial splitter

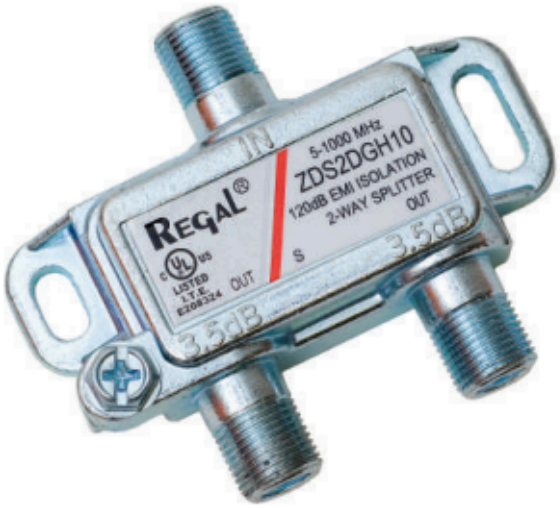


Figure 2-15
Barrel connector

Table 2-1
Coaxial Cables

Rating	Ohms	Use	Connector
RG-59	75	Cable TV	F Type
RG-6	75	Cable TV	F Type

Twinaxial

Twinaxial is a type of cable that contains two central copper conductors wrapped around a single shield (Figure 2-16). You'll see it as a substitute for short fiber connections, generally between equipment within a rack, like switches. For such uses, it's substantially cheaper than

fiber and associated hardware. Twinaxial cable used this way is called a *direct attached cable (DAC)*.

Figure 2-16
Twinaxial cable

Twisted Pair

The most common type of cabling used in networks consists of twisted pairs of cables, bundled together into a common jacket. Each pair in the cable works as a team either transmitting or receiving data. Using a pair of twisted wires rather than a single wire to send a signal reduces a specific type of interference, called *crosstalk*. The more twists per foot, the less crosstalk. Two types of twisted-pair cabling are manufactured: shielded and unshielded.

Shielded Twisted Pair

*Shielded twisted pair (STP) c*onsists of twisted pairs of wires surrounded by shielding to protect them from EMI. There are six types, differentiated by which part gets shielding, such as the whole cable or individual pairs within the cable. Table 2-2 describes the six types. Figure 2-17 shows a typical piece of STP with the cladding partly removed so you can see the internal wiring.

Name	Description
F/UTP	Foil shields the entire cable; inside, the wires are just like UTP.
S/UTP	A braid screen shields the entire cable; inside, the wires are just like UTP.
SF/UTP	A braid screen and foil shield the entire cable; the wires inside are just like UTP.
S/FTP	A braid screen shields the entire cable; foil shields each wire pair inside.
F/FTP	A foil screen shields the entire cable; foil shields each wire pair inside.
U/FTP	No overall shielding; each pair inside is shielded with foil screens.

Table 2-2 STP Standards

Figure 2-17 Shielded twisted pair

EXAM TIP You don't need to memorize the STP variations for the CompTIA Network+ exam. You will, however, see them in the field once you become a network tech. The typical scenario in which you'd deploy UTP is in high-EMI environments, where troubleshooting revealed that the unshielded cable couldn't handle the noise.

Unshielded Twisted Pair *Unshielded Twisted Pair (UTP)* consists of twisted pairs of wires surrounded by a plastic jacket (Figure 2-18). This jacket does not provide any protection from EMI, just a slightly protective skin, so when installing UTP cabling, you must be careful to avoid interference from fluorescent lights, motors, and so forth. UTP costs much less than STP but, in most environments, performs just as well.

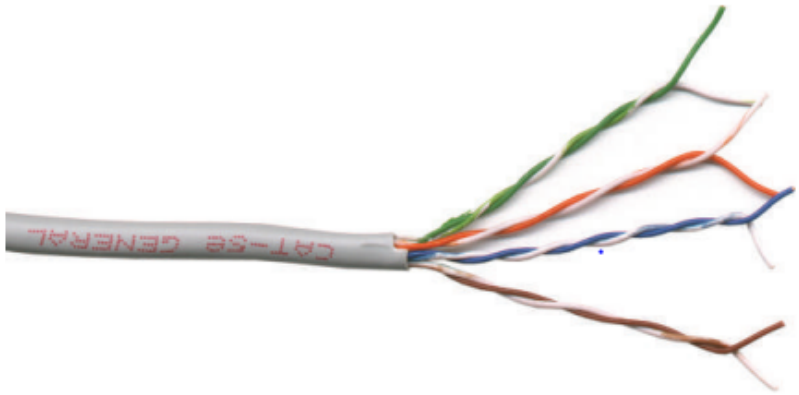


Figure 2-18 Unshielded twisted pair

Twisted-pair cabling has been around since the 1970s and evolving technologies demanded higher speeds. Over the years, manufacturers increased the number of twists per foot, used higher gauge cable, and added shielding to make twisted pair able to handle higher data speeds. To help network installers get the right cable for the right network technology, the cabling industry developed a variety of grades called *category (Cat)* ratings. Cat ratings are officially rated in *megahertz (Mhz)*, indicating the highest frequency the cable can handle. Table 2-3 shows the most common categories along with their status with the TIA (see the subsequent Note for more information).

their status with the TIA (see the subsequent Note for more information).

Cat Rating	Max Frequency	Max Bandwidth	Status with TIA
Cat 3	16 MHz	16 Mbps	Recognized
Cat 4	20 MHz	20 Mbps	No longer recognized
Cat 5	100 MHz	100 Mbps	No longer recognized
Cat 5e	100 MHz	1 Gbps	Recognized
Cat 6[1]	250 MHz	10 Gbps	Recognized
Cat 6a[2]	500 MHz	10 Gbps	Recognized
Cat 7	600 MHz	10+ Gbps	Not recognized
Cat 7a[3]	1000 MHz	40–100 Gbps	Not recognized
Cat 8	2000 MHz	25–40 Gbps	Not recognized

[1] Cat 6 cables can use the full 100-meter length when used with 10/100/1000BASE-T networks. With 10GBASE-T networks, Cat 6 is limited to 55 meters.
[2] Cat 6a cables can use the full 100-meter length with networks up to 10GBASE-T.
[3] Cat 7a cables can theoretically support 40 Gbps at 50 meters; 100 Gbps at 15 meters.

Table 2-3 Cat Ratings for Twisted Pair

Table 2-3 Cat Ratings for Twisted Pair

NOTE Several international groups set the standard for cabling and networking in general. Ready for alphabet soup? At or near the top is the International Organization for

Standardization (ISO). The American National Standards Institute (ANSI) is both the official U.S. representative to ISO and a major international player. ANSI checks the standards and accredits other groups, such as the Telecommunications Industry Association (TIA).

UTP cables handle a certain frequency or cycles per second, such as 100 MHz or 1000 MHz. You could take the frequency number in the early days of networking and translate that into the maximum throughput for a cable. Each cycle per second (or hertz) basically accounted for one bit of data per second. A 10 million cycle per second (10 MHz) cable, for example, could handle 10 million bits per second (10 Mbps). The maximum amount of data that goes through the cable per second is called the *bandwidth*.

EXAM TIP The CompTIA Network+ exam is only interested in your knowledge of Cat 5, Cat 5e, Cat 6, Cat 6a, Cat 7, and Cat 8 cables. (In the field you'll see category represented lowercase and uppercase, so Cat 6a or CAT 6a.)

For current networks, developers have implemented *bandwidth-efficient encoding schemes*, which means they can squeeze more bits into the same signal as long as the cable can handle it. Thus, the Cat 5e cable can handle a throughput of up to 1000 Mbps, even though it's rated to handle a frequency of only up to 100 MHz. Because most networks can run at speeds of up to 1000 Mbps, most new cabling installations use Category 6 (Cat 6) cabling, although a large number of installations use Cat 6a or Cat 7 to future-proof the network. Make sure you can look at twisted cable and know its Cat rating. There are two places to look. First, twisted pair is typically sold in boxed reels, and the manufacturer will clearly mark the Cat level on the box (Figure 2-19). Second, look on the cable itself. The category level of a piece of cable is usually printed on the cable (Figure 2-20).

category level of a piece of cable is usually printed on the cable (Figure 2-20).

Figure 2-19
Cat level marked on box of twisted-pair cabling

Figure 2-19 Cat level marked on box of twisted-pair cabling.

Figure 2-20 Cat level on twisted-pair cabling

The old landline telephones plugged in with a *registered jack (RJ) connector*. Telephones used *RJ-11* connectors, designed to support up to two pairs of UTP wires. Current wired networks use the four-pair *8 position 8 contact (8P8C)* connectors at most techs (erroneously) refer to as RJ-45 connectors (Figure 2-21). (There was an RJ45S connector used in telephones with slightly different keying. They look very similar to the 8P8C connectors, though, so speculation is that the name carried over from technicians installing that new UTP cabling.)

Figure 2-21 RJ-11 (left) and 8P8C/"RJ-45" (right) connectors

EXAM TIP CompTIA follows the common usage for networking cable connectors. You will *not* see 8P8C on the exam; you will *only* see RJ-45.

EXAM TIP CompTIA follows the common usage for networking cable connectors. You will *not* see 8P8C on the exam; you will *only* see RJ-45.

Fiber Optic Cabling and Connectors

Fiber-optic cable transmits light rather than electricity, making it attractive for both highEMI areas and long-distance transmissions. Whereas a single copper cable cannot carry data more than a few hundred meters at best, a single piece of fiber-optic cabling will operate, depending on the implementation, for distances of up to tens of kilometers. A fiber-optic cable has four components: the glass fiber itself (the *core*); the *cladding*, which is the part that makes the light reflect down the fiber; *buffer* material to give strength; and the *insulating jacket* (Figure 2-22).

Figure 2-22 Cross section of fiber-optic cabling

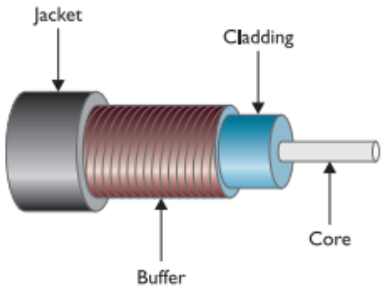


Figure 2-22 Cross section of fiber-optic cabling

You might see the term *fiber cables* on the CompTIA Network+ exam to describe the two varieties of fiber-optic cables discussed in this section. Just as copper cables don't have copper connectors, fiber cables don't have *fiber connectors*, but that's the term used in the CompTIA Network+ Spare Parts list. I'll discuss cables and connector types shortly.

NOTE

For those of you unfamiliar with it, the odd little u-shaped symbol describing fiber cable size (µ) stands for micro, or 1/1,000,000.

Fiber-optic cabling is manufactured with many different diameters of core and cladding. Cable manufacturers use a two-number designator to define fiber-optic cables according to their core and cladding measurements. Common fiber-optic cable sizes are 9/125 µµm, 50/125 µµm, and 62.5/125 µµm. Almost all network technologies that use fiber-optic cable require pairs of fibers. One fiber is used for sending, the other for receiving. In response to the demand for two-pair cabling, manufacturers often connect two fibers together to create *duplex* fiber-optic cabling (Figure 2-23).

Figure 2-23
Duplex fiber-optic cable

Light can be sent down a fiber-optic cable as regular light or as laser light. The two types of light require totally different fiber-optic cables. Network technologies that use fiber optics use LEDs (light emitting diodes) to send light signals. A fiber-optic cable that uses LEDs is known as *multimode fiber (MMF)*.

MMF can use a form of laser called a *vertical-cavity surface-emitting laser (VCSEL)*, which differs substantially from the lasers used in SMF. You'll find VCSELs in computer mice and laser printers, among other uses. And in case you're curious, VCSELs are not on the CompTIA Network+ exam.

A fiber-optic cable that uses lasers is known as *single-mode fiber (SMF)*. Using laser light and single-mode fiber-optic cables prevents a problem unique to multimode fiber optics called *modal distortion* (signals sent at the same time don't arrive at the same time because the paths differ slightly in length) and enables a network to achieve phenomenally high transfer rates over incredibly long distances.

NOTE A *nano* – abbreviated as *n* – stands for 1/1,000,000,000, or one-billionth of whatever. Here you'll see it as a nanometer (nm), one-billionth of a meter. That's one tiny wavelength!

Fiber optics also defines the wavelength of light used, measured in nanometers (nm). Multimode cables transmit 850-nm or 1300-nm wavelengths, whereas single-mode transmits either 1310 nm or 1550 nm, depending on the laser.

NOTE The ANSI/TIA-568.3-D standard defines the nomenclature for fiber. The multimode standard prefix is OM (single-mode is OS). OM1 and OM2 are used in shorter runs with LEDs. OM3, OM4, and OM5 can use lasers and run at higher bandwidths, thus providing faster speed and greater distances. The ANSI/TIA 598-C standard provides guidelines for color-coding

various fiber types. Single-mode fiber is yellow. OM1 and OM2 are both orange. OM3 and OM4 sport aqua, and OM5 appears in a spectacular lime green.

Fiber-optic cables come in a broad choice of connector types. There are over one hundred different connectors, but the four you need to know for the CompTIA Network+ exam are ST, SC, LC, and MT-RJ. Figure 2-24 shows the first three; Figure 2-25 shows an MT-RJ connector.

Figure 2-24 From left to right: ST, SC, and LC fiber-optic connectors

Figure 2-25 MT-RJ fiber-optic connector

Although all fiber connectors must be installed in pairs, the ST and SC connectors traditionally have unique ends. The LC and MT-RJ connectors are always duplex, meaning both the send and receive cables are attached. You can certainly find SC connectors or sleeves to make them duplex too, so don't get too caught up with which can be which. We'll revisit fiber-optic connectors in Chapter 4 when we discuss implementation of specific networking standards.

NOTE Most technicians call common fiber-optic connectors by their initials— such as ST, SC, or LC—perhaps because there's no consensus about what words go with those initials. ST probably stands for *straight tip*, although others call it *snap and twist*. But SC and LC? How about *subscriber connector*, *standard connector,* or *Siemon connector* for the former, and *local connector* or *Lucent connector* for the latter? If you want to remember the connectors for the exam, try these: *snap and twist* for the bayonet-style ST connectors; *stick and click* for the straight push-in SC connectors; and *little connector* for the … little … LC connector.

Fire Ratings

Did you ever see the movie *The Towering Inferno?* Don't worry if you missed it— *The Towering Inferno* was one of the better disaster movies of the 1970s, although it was no *Airplane!* Anyway,

Steve McQueen stars as the fireman who saves the day when a skyscraper goes up in flames because of poor-quality electrical cabling. The burning insulation on the wires ultimately spreads the fire to every part of the building. Although no cables made today contain truly flammable insulation, the insulation is made from plastic, and if you get any plastic hot enough, it will create smoke and noxious fumes. The risk of burning insulation isn't fire—it's smoke and fumes. To reduce the risk of your network cables burning and creating noxious fumes and smoke, Underwriters Laboratories and the National Electrical Code (NEC) joined forces to develop cabling *fire ratings*. The two most common fire ratings are PVC and plenum. Cable with a *polyvinyl chloride (PVC) rating* has no significant fire protection. If you burn a *PVC-rated cable*, it creates lots of smoke and noxious fumes. Burning *plenum-rated cable* creates much less smoke and fumes, but plenum-rated cable costs about three to five times as much as PVC-rated cable. Most city ordinances require the use of plenum cable for network installations. The bottom line in such scenarios? Get plenum! The space between the acoustical tile ceiling in an office building and the actual concrete ceiling above is called the *plenum* – hence the name for the proper fire rating of cabling to use in that space. A third type of fire rating, known as *riser*, designates the proper cabling to use for vertical runs between floors of a building. *Riser-rated cable* provides less protection than plenum cable, though, so most installations today use plenum for runs between floors.

EXAM TIP Look for a troubleshooting scenario question on the CompTIA Network+ exam that asks you to compare plenum versus PVC cable best use. If it goes in the wall, make it plenum!

Networking Industry Standards - IEEE

The *Institute of Electrical and Electronics Engineers (IEEE)* defines industry-wide standards that promote the use and implementation of technology. In February 1980, a committee called the 802 Working Group took over from the private sector the job of defining network standards. The IEEE 802 committee defines frames, speeds, distances, and types of cabling to use in a network environment. Concentrating on cables, the IEEE recognizes that no single cabling solution can work in all situations and, therefore, provides a variety of cabling standards.

IEEE committees define standards for a wide variety of electronics. The names of these committees are often used to refer to the standards they publish. The IEEE 1284 committee, for example, set standards for parallel communication, so you would see parallel cables marked "IEEE 1284 compliant," as in Figure 2-26.

Figure 2-26 Parallel cable marked IEEE 1284 compliant

The IEEE 802 committee sets the standards for networking. Although the original plan was to define a single, universal standard for networking, it quickly became apparent that no single solution would work for all needs. The 802 committee split into smaller subcommittees, with names such as IEEE 802.3 and IEEE 802.11. Table 2.4 shows the currently recognized IEEE 802 subcommittees and their areas of jurisdiction. The missing numbers, such as 802.2 and 802.12, were used for committees long-ago disbanded. Each subcommittee is officially called a Working Group, except the few listed as a Technical Advisory Group (TAG) in the table.

IEEE 802.1	Higher Layer LAN Protocols (with many subcommittees, like 802.1X for port-based network access control)
IEEE 802.3	Ethernet (with a ton of subcommittees, such as 802.3ae for 10-Gigabit Ethernet)
IEEE 802.11	Wireless LAN (WLAN); specifications, such as Wi-Fi, and many subcommittees
IEEE 802.15	Wireless Personal Area Network (WPAN)
IEEE 802.18	Radio Regulatory Technical Advisory Group
IEEE 802.19	Wireless Coexistence Working Group
IEEE 802.24	Vertical Applications Technical Advisory Group

Table 2-4 Some IEEE 802 Subcommittees

Table 2-4 Some IEEE 802 Subcommittees

Chapter Review

Questions

1. Which of the following topologies required termination?
    a. Star
    b. Bus
    c. Mesh
    d. Ring
2. Star-bus is an example of an _____ topology.

a. Transitional
b. System
c. Hybrid
d. rampant

3. Of the topologies listed, which of the following is the most fault-tolerant?
    a. Point-to-point
    b. Bus
    c. Star
    d. Ring
4. What term is used to describe the interconnectivity of network components?
    a. Segmentation
    b. Map
    c. Topology
    d. Protocol
5. Coaxial cables all have a(n) _____ rating.
    a. resistance
    b. Watt
    c. speed
    d. Ohm
6. Which of the following is a type of coaxial cable?
    a. RJ-45
    b. RJ-6
    c. BNC
    d. Barrel
7. Which network topology connected nodes with a ring of cable?
    a. Star
    b. Bus
    c. Ring
    d. Mesh
8. Which network topology is most commonly seen only in wireless networks?
    a. Star
    b. Bus
    c. Ring
    d. Mesh
9. Which of the following is a duplex fiber-optic connection?
    a. LC
    b. RJ-45
    c. ST
    d. SC
10. Which of the following is the most common category of UTP used in new cabling installations?
    a. Cat 3
    b. Cat 5e
    c. Cat 6

d. Cat 7

Answers

1. B. In a bus topology, all computers connected to the network via a main line. The cable had to be terminated at both ends to prevent signal reflection. 2. C. Star-bus is a hybrid topology because it uses a star physical topology and a bus signal topology. 3. C. Of the choices listed, only star topology has any fault tolerance. 4. C. Topology is the term used to describe the interconnectivity of network components. 5. D. All coaxial cables have an Ohm rating. RG-59 and RG-6 both are rated at 75 Ohms. 6. B. RG-6 is a type of coaxial cable. 7. C. The aptly named ring topology connected nodes with a central ring of cable. 8. D. Mesh is, for the most part, unique to wireless networks. 9. A. Of the options given, only the LC connector is designed for duplex fiber-optic. 10. C. Cat 6 is the most common cabling category installed today, although Cat 6a and Cat 7 are gaining in popularity.

Chapter 3 - Ethernet Basics

The CompTIA Network+ certification exam expects you to know how to:

- 1.3 Summarize the types of cables and connectors and explain which is the appropriate type for a solution
- 2.1 Compare and contrast various devices, their features, and their appropriate placement on the network
- 2.3 Given a scenario, configure and deploy common Ethernet switching features
- 5.2 Given a scenario, troubleshoot common cable connectivity issues and select the appropriate tools

To achieve these goals, you must be able to

- Define and describe Ethernet
- Explain early Ethernet implementations
- Describe ways to enhance and extend Ethernet networks

In the beginning, there were no networks. Computers were isolated, solitary islands of information in a teeming sea of proto-geeks who banged out binary messages with wooden clubs and wore fur pocket protectors. Okay, maybe it wasn't that bad, but if you wanted to move a file from one machine to another, you had to use Sneakernet, which meant you saved the file on a disk, laced up your tennis shoes, and hiked over to the other system. All that walking no doubt produced lots of health benefits, but frankly, proto-geeks weren't all that into health benefits—they were into speed, power, and technological coolness in general. (Sound familiar?) It's no wonder, then, that geeks everywhere agreed on the need to replace walking with some form of networking technology that connects computers together to transfer data at very high speeds. This chapter explores the networking technology that eventually took control of the

industry, Ethernet. We'll start with basic terminology, then look at two early forms of Ethernet. The chapter finishes with a discussion on enhancing and expanding Ethernet networks.

Historical/Conceptual

In 1973, Xerox answered the challenge of moving data without sneakers by developing *Ethernet*, a networking technology standard based on a bus topology. The original Ethernet used a single piece of coaxial cable to connect several computers, enabling them to transfer data at a rate of up to three million bits per second (3 Mbps). Although slow by today's standards, this early version of Ethernet was a huge improvement over manual transfer methods and served as the foundation for all later versions of Ethernet. Ethernet remained a largely in-house technology within Xerox until 1979, when Xerox decided to look for partners to help promote Ethernet as an industry standard. Xerox worked with Digital Equipment Corporation (DEC) and Intel to publish what became known as the Digital/Intel/Xerox (DIX) standard. The DIX Ethernet standard improved on the original Ethernet standard, increasing speed to a screaming 10 Mbps. These companies then did something visionary: they transferred (one might also say gave away) control of the Ethernet standard to the Institute of Electrical and Electronics Engineers (IEEE), which, in turn, created the *802.3 (Ethernet)* working group that continues to control the Ethernet standard to this day. By transferring control to IEEE, Ethernet became an open standard, enabling anyone to make interchangeable Ethernet equipment. Making Ethernet an open standard made Ethernet much cheaper than any alternative technology and certainly contributed to Ethernet winning the marketplace.

802.3 Standards

The 802.3 working group defines wired network standards that share the same basic frame type and network access method. Each of these variants is under the IEEE 802.3 standard, each with its own identifier. Here's a small selection of 802.3 standards:

- **802.3i** 10 Mbps Ethernet using twisted pair cabling (1990)
- **802.3ab** Gigabit Ethernet over twisted pair (1999)
- **802.3by** 25 Gigabit Ethernet over fiber (2016)
- **802.3cm** 400 Gigabit Ethernet over multimode fiber (2020)
- **802.3cu** 100 Gigabit and 400 Gigabit Ethernet over single mode fiber using 100 Gbps lanes (2021)

Because the technologies share essential components, you can communicate among them just fine. The implementation of the network might be different, but the frames remain the same. Ethernet's designers faced the same challenges as the designers of any network: how to send data across the wire, how to identify the sending and receiving computers, and how to determine which computer should use the shared cable at what time. The engineers resolved these issues by using data frames that contain MAC addresses to identify computers on the network and by using a process called CSMA/CD (discussed shortly) to determine which

machine should access the wire at any given time. You saw some of this in action in Chapter 1, but now I need to introduce you to a bunch of additional terms.

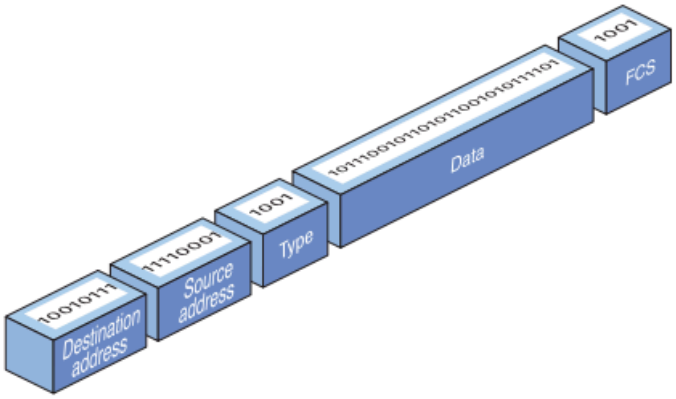
NOTE The source for all things Ethernet is but a short click away on the Internet. For starters, check out www.ieee802.org.

Test Specific - Ethernet Frames

All network technologies break data transmitted between computers into segments or datagrams, placed into packets that in turn get placed into *frames*, as you'll recall from Chapter 1. Using frames addresses two networking issues. First, frames prevent any single machine from monopolizing the shared bus cable. Second, they make the process of retransmitting lost data more efficient.

The process you saw in Chapter 1 of transferring a word processing document between two computers illustrates these two issues. First, if the sending computer sends the document as a single huge frame, the frame will monopolize the cable and prevent other machines from using the cable until the entire file gets to the receiving system. Using relatively small frames enables computers to share the cable easily—each computer listens on the *network segment*, sending a few frames of data whenever it detects that no other computer is transmitting. Second, in the real world, bad things can happen to good data. When errors occur during transmission, the sending system must retransmit the frames that failed to reach the receiving system in good shape. If a word processing document were transmitted as a single massive frame, the sending system would have to retransmit the entire frame—in this case, the entire document. Breaking the file up into smaller frames enables the sending computer to retransmit only the damaged frames. Because of these benefits—shared access and more efficient retransmission—all networking technologies use frames. In Chapter 1, you saw a generic frame. Let's take what you know of frames and expand on that knowledge by inspecting the details of an Ethernet frame. A basic Ethernet frame contains five fields: the *destination address* – the MAC address of the frame's recipient; the *source address* – the MAC address of the frame's recipient; the *source address* – the MAC address of the sending system; the *type* of the data; the data itself; and a *frame check sequence*. Figure 3-1 shows these components. Transmission of a frame starts with a *preamble* and can also include some extra filler called a *pad*. Let's look at each piece.

Figure 3-1
Ethernet frame



Preamble

A *preamble,* a 7-byte series of alternating ones and zeroes followed by a 1-byte *start frame delimiter* or an 8-byte series of alternating ones and zeroes, always precedes a frame. The preamble gives a receiving NIC time to realize a frame is coming and to know exactly where the frame starts. The preamble is added by the sending NIC.

MAC Addresses

Each NIC on an Ethernet network must have a unique identifying address. Ethernet identifies the NICs on a network using special 48-bit (6-byte) binary addresses known as *MAC addresses.*

EXAM TIP

The CompTIA Network+ exam might describe MAC addresses as 48-bit binary addresses or 6-byte binary addresses.

In a bus network, all the connected computers could see all traffic. The *destination address* in the frame enabled NICs to examine each frame and process only frames intended for them. The *source address* in the frame enabled the recipient to respond accurately.

Type

An Ethernet frame may carry one of several types of data. The *type* field helps the receiving computer interpret the frame contents at a very basic level. This way the receiving computer can tell if the frame contains IPv4 data, for example, or IPv6 data. (See Chapter 6 for more details on IPv4; I cover IPv6 in Chapter 12.)

The type field does *not* tell you if the frame carries higher-level data, such as an e-mail message or Web page. You have to dig deeper into the data section of the frame to find that information.

Data

The *data* part of the frame contains whatever payload the frame carries. If the frame carries an IP packet, that packet will include extra information, such as the IP addresses of the source and destination systems.

Pad

The minimum Ethernet frame is 64 bytes in size, but not all of that has to be actual data. If an Ethernet frame has fewer than 64 bytes of data to haul, the sending NIC automatically adds extra data—a *pad* – to bring the data up to the minimum 64 bytes. Padding is used all the time in modern networking.

Frame Check Sequence

The *frame check sequence (FCS)* enables Ethernet nodes to recognize when bad things happen to good data. Machines on a network must be able to detect when data has been damaged in transit. To detect errors, the computers on an Ethernet network attach a special code to each frame. When creating an Ethernet frame, the sending machine runs the data through a special mathematical formula called a *cyclic redundancy check (CRC)* and attaches the result, the FCS, to the frame as the trailer. The receiving machine opens the frame, performs the same calculation, and compares its answer with the one included with the frame. If the answers do not match, the receiving machine drops the frame.

Early Ethernet Standards

Contemplating the physical network brings up numerous questions. What kind of cables should you use? What should they be made of? How long can they be? For these answers, turn to the IEEE 802.3 standard, both true bus and star-bus versions.

Bus Ethernet

The original Ethernet networks employed a true bus topology, meaning every computer on a network connected to the same cable, the bus. Every version of Ethernet invented since the early 1990s uses a hybrid star-bus topology. At the center of these early networks was a *hub*. A hub was nothing more than an electronic *repeater* – it interpreted the ones and zeroes coming in from one port and repeated the same signal out to the other connected ports. Hubs did not send the same signal back down the port that originally sent it (Figure 3-2). Any scenario involving these early networks found the placement of a hub at the center of the network.

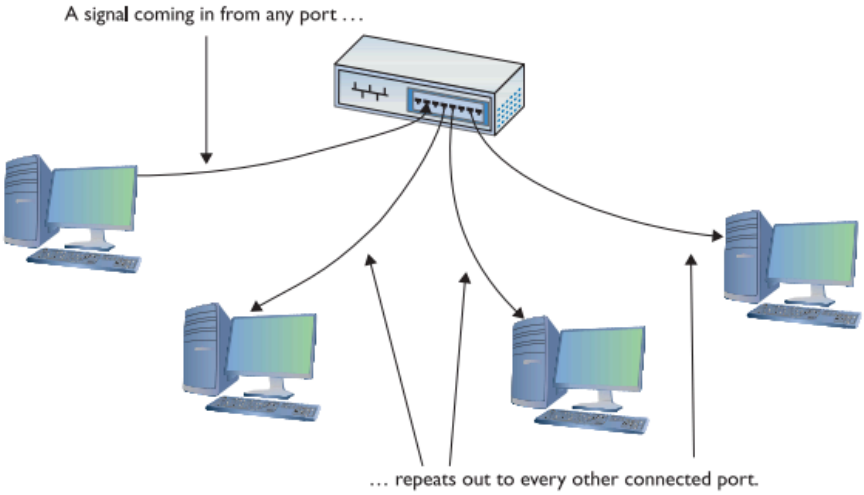


Figure 3-2 Ethernet hub

10BASE-T

In 1990, the IEEE 802.3 committee created a version of Ethernet called 10BASE-T that rapidly became the most popular network technology in the world, replacing competing and now long-gone competitors with names like Token Ring and LocalTalk. The classic 10BASE-T network consisted of two or more computers connected to a central hub. The NICs connected with wires as specified by the 802.3 committee. The name 10BASE-T follows roughly the same naming convention used for earlier Ethernet cabling systems. The number 10 refers to the speed: 10 Mbps. The word BASE refers to the signaling type: baseband. (*Baseband* means that the cable carries only one signal. Contrast this with *broadband* – as in cable television – where the cable carries multiple signals or channels.) The letter T refers to the type of cable used: twisted pair. 10BASE-T used unshielded twisted pair (UTP) cabling.

UTP

Officially, 10BASE-T required the use of Cat 3 (or higher), two-pair, UTP cable. One pair of wires sent data to the hub while the other pair received data from the hub. Even though 10BASE-T only required two-pair cabling, everyone installed four-pair cabling to connect devices to the hub as insurance against the possible requirements of newer types of networking (Figure 3-3). Not surprisingly, this came in handy very soon. See Chapter 4 for more details. Most UTP cables (then and now) come with stranded Kevlar fibers to give the cable added strength, which, in turn, enables installers to pull on the cable without excessive risk of literally ripping it apart.

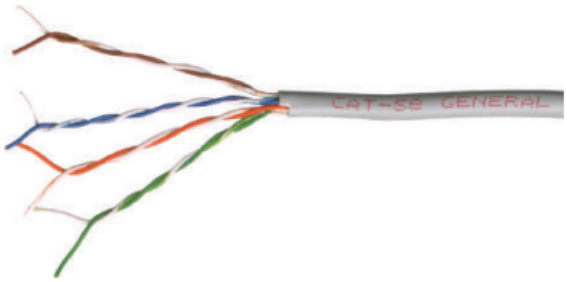


Figure 3-3 A typical four-pair Cat 5e unshielded twisted pair cable

10BASE-T also introduced the networking world to the *RJ-45 connector* (Figure 3-4). Each pin on the RJ-45 connects to a single wire inside the cable; this enables devices to put voltage on the individual wires within the cable. The pins on the RJ-45 are numbered from 1 to 8, as shown in Figure 3-5.

Figure 3-4 Two views of an RJ-45 connector

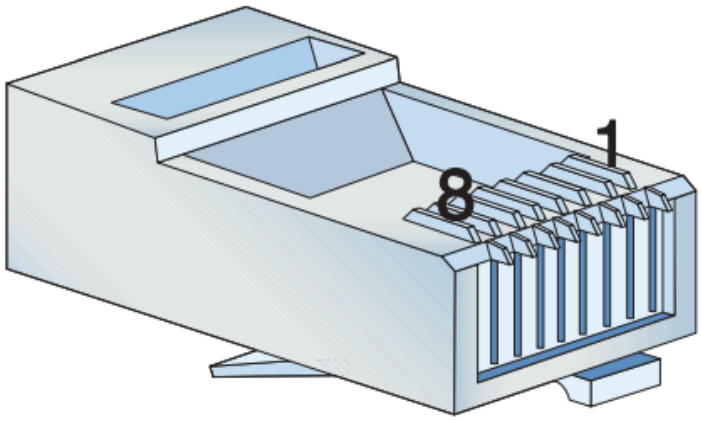


Figure 3-5 The pins on an RJ-45 connector are numbered 1 through 8.

The 10BASE-T standard designates some of these numbered wires for specific purposes. As mentioned earlier, although the cable has four pairs, 10BASE-T used only two of the pairs. 10BASE-T devices used pins 1 and 2 to send data, and pins 3 and 6 to receive data. Even though one pair of wires sent data and another received data, a 10BASE-T device that was connected to a hub could not send and receive simultaneously. See “CSMA/CD” later in this chapter for details about collisions and using a shared bus. NICs that can communicate in only one direction at a time run in half-duplex mode. Later advances (as you’ll see shortly) enabled NICs to send and receive at the same time, thus running in *full-duplex* mode.

An RJ-45 connector is sometimes called a *crimp*, and the act (some folks call it an art) of installing a crimp onto the end of a piece of UTP cable is called *crimping*. The tool used to secure a crimp onto the end of a cable is a crimper. Each wire inside a UTP cable must connect to the proper pin inside the crimp. Manufacturers color-code each wire within a piece of four-pair UTP to assist in properly matching the ends. Each pair of wires consists of a solid-colored wire and a striped wire: blue/blue-white, orange/orange-white, brown/brown-white, and green/green-white (Figure 3-6).

Figure 3-6 Color-coded pairs (note the alternating solid and striped wires)

NOTE As noted in Chapter 2, the real name for RJ-45 is *8 position 8 contact (8P8C)* modular plug. The term RJ-45 is so prevalent, however, that nobody but the nerdiest of nerds calls it by its real name. Stick to RJ-45.

The Telecommunications Industry Association/Electronics Industries Alliance (TIA/ EIA) defines the industry *termination standard* for correct crimping of four-pair UTP. You'll find two standards mentioned on the CompTIA Network+ exam: *TIA/EIA 568A* and *TIA/EIA 568B*. Figure 3-7 shows the TIA/EIA 568A and TIA/EIA 568B color-code standards. Note that the wire pairs used by 10BASE-T (1 and 2, 3 and 6) come from the same color pairs (green/green-white and orange/orange-white). Following an established color-code scheme, such as TIA/EIA 568A, ensures that the wires match up correctly at each end of the cable.

each end of the cable.

Figure 3-7 The TIA/EIA 568A and 568B standards

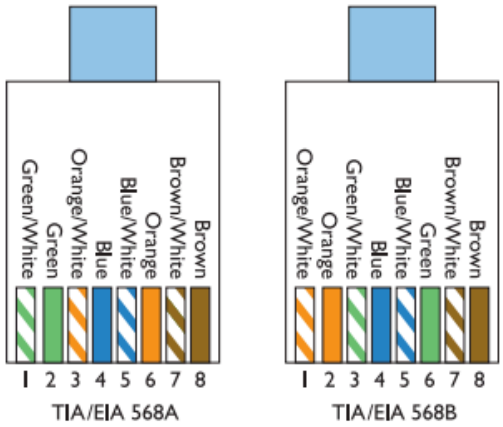


Figure 3-7 The TIA/EIA 568A and 568B standards

EXAM TIP

The current twisted pair cabling standard includes the same wiring standards as TIA/EIA 568A and TIA/EIA 568B. It's all just wrapped up in a slightly different name: *ANSI-TIA-568-D*. When the EIA left the planet in 2011, the names of the standards changed. CompTIA continues to use the older names on exams.

The ability to make your own Ethernet cables is a plus for a network tech. With a reel of Cat 5e, a bag of RJ-45 connectors, a moderate investment in a crimping tool, and a little practice, you can kiss those mass-produced cables goodbye! You can make cables to your own length specifications, replace broken RJ-45 connectors that would otherwise mean tossing an entire cable, and, in the process, save your company or clients time and money.

EXAM TIP

An easy trick to remembering the difference between 568A and 568B is the word “GO.” The green and orange pairs are swapped between 568A and 568B, whereas the blue and brown pairs stay in the same place! For the CompTIA Network+ exam, you will be tested on the TIA/EIA 568A or 568B color codes. Memorize them. You’ll see the standards listed as EIA/TIA 568A, TIA/EIA568A, T568A, or just 568A. Know the A and B and you’ll be fine.

10BASE-T Limits and Specifications

Like any other Ethernet standard, 10BASE-T had limitations, both on cable distance and on the number of computers. The key distance limitation for 10BASE-T was the distance between the hub and the computer. The twisted pair cable connecting a computer to the hub could not exceed 100 meters in length. A 10BASE-T hub could connect no more than 1024 computers, although that limitation rarely came into play. It made no sense for vendors to build hubs that large—or more to the point, that *expensive*.

10BASE-T Summary

- **Speed** 10 Mbps
- **Signal type** Baseband
- **Distance** 100 meters between the hub and the node
- **Node limit** No more than 1024 nodes per hub
- **Topology** Star-bus topology: physical star, logical bus
- **Cable type** Cat 3 or better UTP cabling with RJ-45 connectors

SIM Check out the Chapter 3 Challenge! sim “T-568B” here: https://totalsem.com/008
It’s a great tool for getting the colors set in your head.

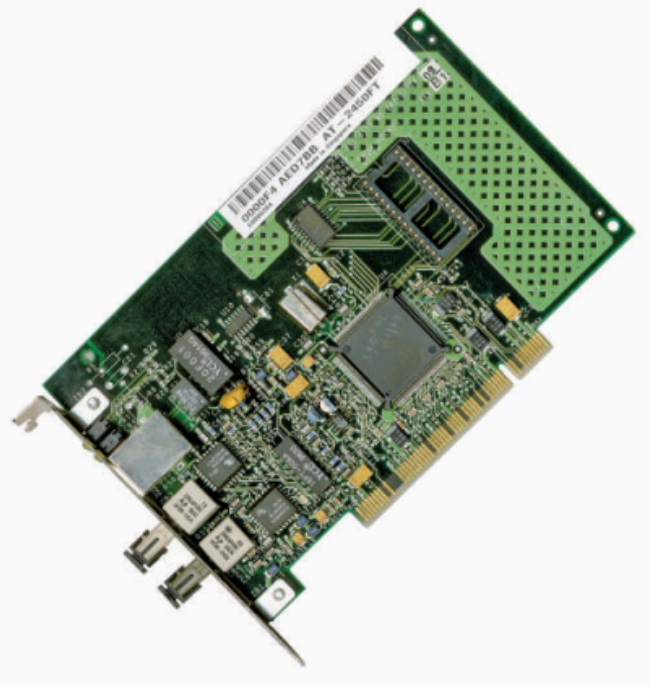
10BASE-FL

Just a few years after the introduction of 10BASE-T, a fiber-optic version, called *10BASE-FL,* appeared. As you know from the previous chapter, fiber-optic cabling transmits data using pulses of light instead of using electrical current. Using light instead of electricity addresses the three key weaknesses of copper cabling. First, optical signals can travel much farther. The maximum length for a 10BASE-FL cable was up to 2 kilometers, depending on how you configured it. Second, fiber-optic cable is immune to electrical interference, making it an ideal choice for high-interference environments. Third, the cable is much more difficult to tap into, making fiber a good choice for environments with security concerns. 10BASE-FL used *multimode* 62.5/125 µm (OM1) fiber-optic cabling and employed either an SC or ST connector.

Figure 3-8 shows a typical 10BASE-FL card. Note that it uses two fiber connectors— one to send and one to receive. All fiber-optic networks use at least two fiber-optic cables. Although

10BASE-FL enjoyed some popularity for a number of years, most networks today are using the same fiber-optic cabling to run far faster network technologies.

Figure 3-8
Typical 10BASE-FL card



10BASE-FL Summary

- **Speed** 10 Mbps
- **Signal type** Baseband
- **Distance** 2000 meters between the hub and the node
- **Node limit** No more than 1024 nodes per hub
- **Topology** Star-bus topology: physical star, logical bus
- **Cable type** Multimode 62.5/125 µm (OM1) fiber-optic cabling with ST or SC connectors

So far you've seen two different flavors of star-bus Ethernet, 10BASE-T and 10BASEFL. Even though these used different cabling and hubs, they used Ethernet frames. As a result, interconnecting flavors of Ethernet were (and still are) common. Because 10BASE-T and 10BASE-FL used different types of cable, you could use a *media converter* (Figure 3-9) to interconnect different Ethernet types.

Figure 3-9
Typical copper-to-fiber Ethernet media converter (photo courtesy of TRENDnet)

Figure 3-9 Typical copper-to-fiber Ethernet media converter (photo courtesy of TRENDnet)

EXAM TIP 10BASE-FL is *not* on the CompTIA Network+ exam. Its successor, 100BASE-FX is on the exam; we'll get there in Chapter 4.

CSMA/CD

One of the issues with bus communication is that devices essentially share the same cable. This applies to pure bus networks and hybrid star-bus networks as well. The NICs need some way to determine which machine should send data at which time. Ethernet designers came up with a clever way to handle the issue of potential collisions. Ethernet networks used a system called *carrier-sense multiple access with collision detection (CSMA/CD)* to determine which computer should use a shared cable at a given moment.

Carrier sense meant that each node using the network examined the cable before sending a data frame (Figure 3-10). If another machine was using the network, the node detected traffic on the segment, waited a few milliseconds, and then rechecked. If it detected no traffic, the node sent out its frame.

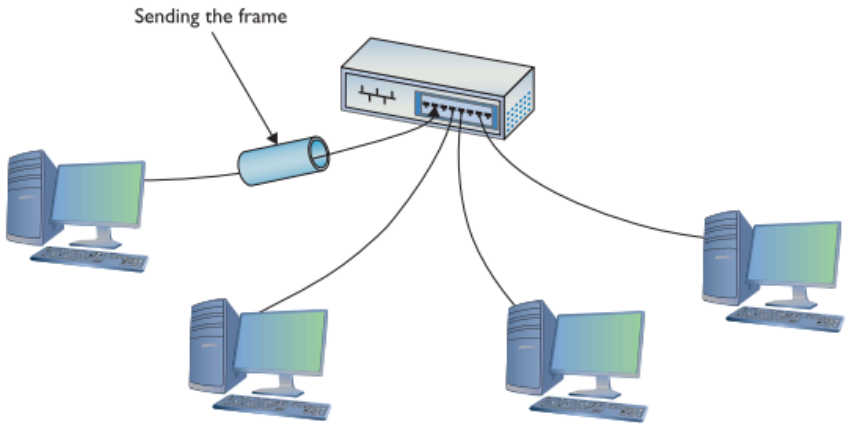


Figure 3-10 No one else is talking—send the frame!

EXAM TIP CSMA/CD was a network access method that mapped to the IEEE 802.3 standard for Ethernet networks. It's disabled in modern full-duplex networks, but still shows up on the objectives for an exam in your near future.

Multiple access meant that all machines had equal access to the wire. If the line was free, any Ethernet node could send a frame. From Ethernet's point of view, it didn't matter what function the node performed: it could have been a desktop system running Windows 98 or a file server running Windows Server or Linux. As far as early Ethernet was concerned, a node was a node was a node and access to the cable was assigned strictly on a first-come, first-served basis. So what happened if two machines, both listening to the cable, simultaneously decided that it was free and tried to send a frame? A collision occurred, and both of the transmissions were lost (Figure 3-11). A collision resembles the effect of two people talking at the same time: the listener hears a mixture of two voices and can't understand either one.

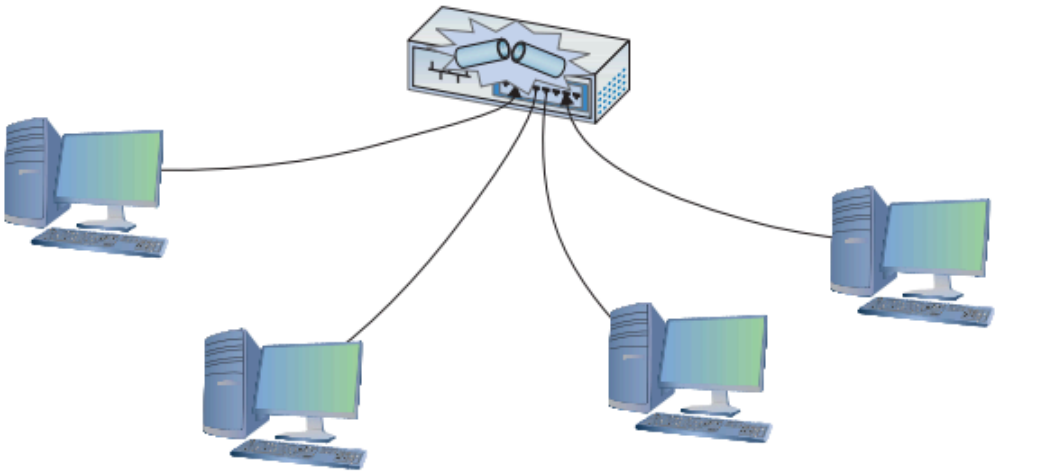


Figure 3-11 Collision!

Figure 3-11 Collision!

When two NICs are sent at the same time, they'd sense the overlapping signals and immediately know that a collision occurred. When they detected a collision, both nodes stopped transmitting. They then each generated a random number to determine how long to wait before trying again. If you imagine that each machine rolled its magic electronic dice and waited for that number of seconds, you wouldn't be too far from the truth, except that the amount of time an Ethernet node waited to retransmit was much shorter than one second (Figure 3-12). Whichever node generated the lowest random number began its retransmission first, winning the competition to use the wire. The losing node then saw traffic on the wire and waited for the wire to be free again before attempting to retransmit its data.

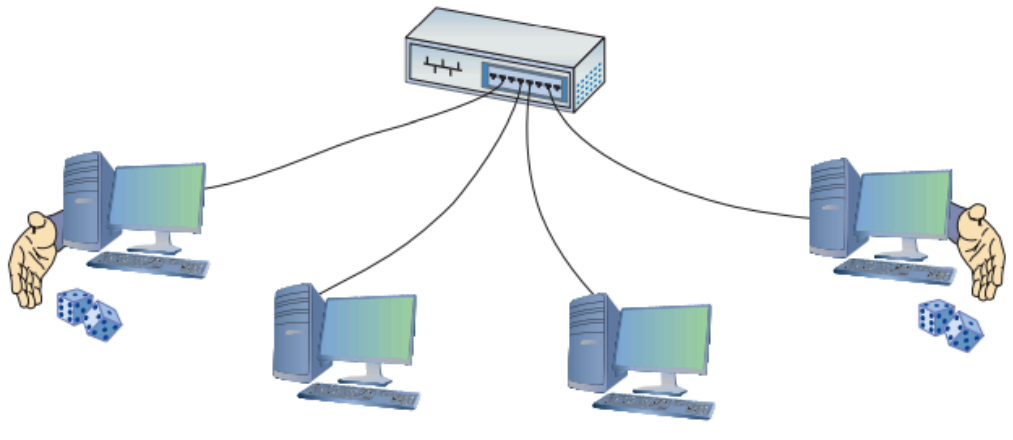


Figure 3-12 Rolling for timing

Figure 3-12 Rolling for timing

Collisions were a normal part of the operation of early Ethernet networks, because every device shared a bus. A group of nodes that have the capability of sending frames at the same time as each other, resulting in collisions, is called a collision domain. Better technology today eliminates collisions.

Enhancing and Extending Ethernet Networks

While plain-vanilla 10BASE-T Ethernet performed well enough for first-generation networks (which did little more than basic file and print sharing), by the early 1990s networks used more-demanding applications, such as Lotus Notes, SAP business management software, and Microsoft Exchange, which quickly saturated a 10BASE-T network. Fortunately, those crazy kids over at the IEEE kept expanding the standard, giving the network tech in the trenches a new tool that provided additional bandwidth—the switch. Additionally, more companies and organizations adopted Ethernet, leading to a demand for larger networks, both geographically and in the number of nodes that could interconnect. Hubs were cranky and creaky; switches brought much better scalability.

The Trouble with Hubs

A classic 10BASE-T network with a hub could only have one message on the wire at any time. When two computers sent at the same time, the hub dutifully repeated both signals. The nodes recognized the collision and, following the rules of CSMA/CD, attempted to resend. Add in enough computers and the number of collisions increased, lowering the effective transmission speed for the whole network. A busy network became a slow network because all the computers shared the same collision domain.

EXAM TIP

Adding another hub or two to an early Ethernet network enabled you to add more devices, but also compounded the problem with collisions. In such a scenario, you could connect networks using a bridge. A *bridge* acted like a repeater to connect two networks, but then went a step further—filtering and forwarding traffic between those segments based on the MAC addresses of the computers on those segments. This placement between two segments preserved bandwidth, making larger Ethernet networks possible. You'll see the term "bridge" applied to modern devices, primarily in wireless networking. The interconnectedness of network segments is similar, but the devices are fundamentally different. See Chapter 14 for the scoop on wireless.

Switches to the Rescue

An Ethernet *switch* looks like a hub, because all nodes plug into it (Figure 3-13). But switches don't function like hubs inside. Switches come with extra smarts that enable them to take advantage of MAC addresses, effectively creating point-to-point connections between two conversing computers. This gives every conversation between two computers the full bandwidth of the network.

Figure 3-13 Hub (top) and switch (bottom) comparison

Figure 3-13 Hub(top) and switch(bottom) comparison

To see a switch in action, check out Figure 3-14. When you first turn on a switch, it acts like a hub, passing all incoming frames right back out to all the other ports. As it forwards all frames, however, the switch copies the source MAC addresses and quickly creates a table of the MAC addresses of each connected computer, called a *media access control (MAC) address table*.

Port	MAC Address
1	None
2	28-4F-C2-31-22-B2
3	None
4	45-9D-84-D2-AA-10
5	F1-E2-A9-9C-41-BC
6	None
7	AD-83-F2-90-D2-36
8	None

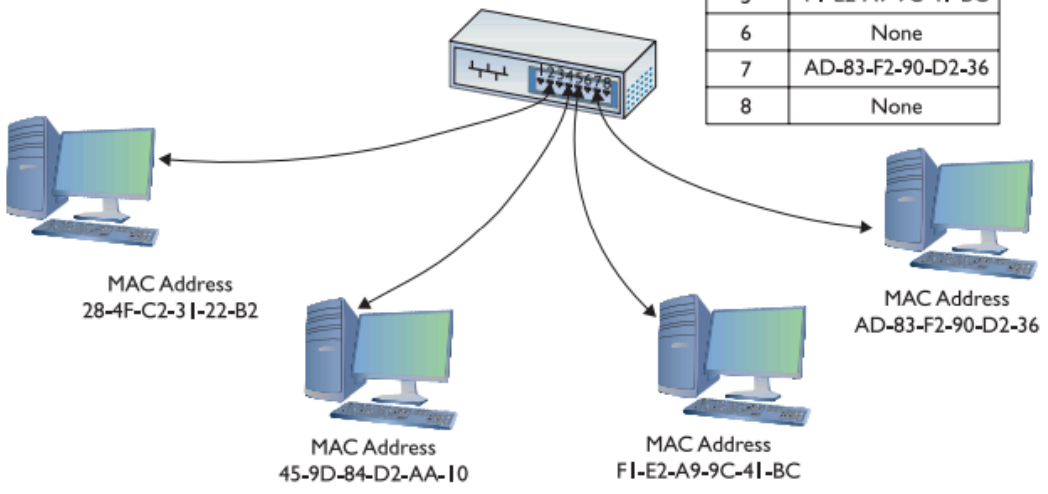


Figure 3-14 A switch tracking MAC addresses

Figure 3-14 A switch tracking MAC addresses

EXAM TIP One classic difference between a hub and a switch is in the repeating of frames during normal use. Although it's true that switches initially forward all frames, they filter by MAC

address once they complete port mapping. Hubs never learned and always forwarded all frames.

As soon as this table is created, the switch begins to do something amazing. When a computer sends a frame into the switch destined for another computer on the same switch, the switch acts like a telephone operator, creating an on-the-fly connection between the two devices. While these two devices communicate, it's as though they are the only two computers on the network. Figure 3-15 shows this in action. Because the switch handles each conversation individually, each conversation runs at the full network speed.

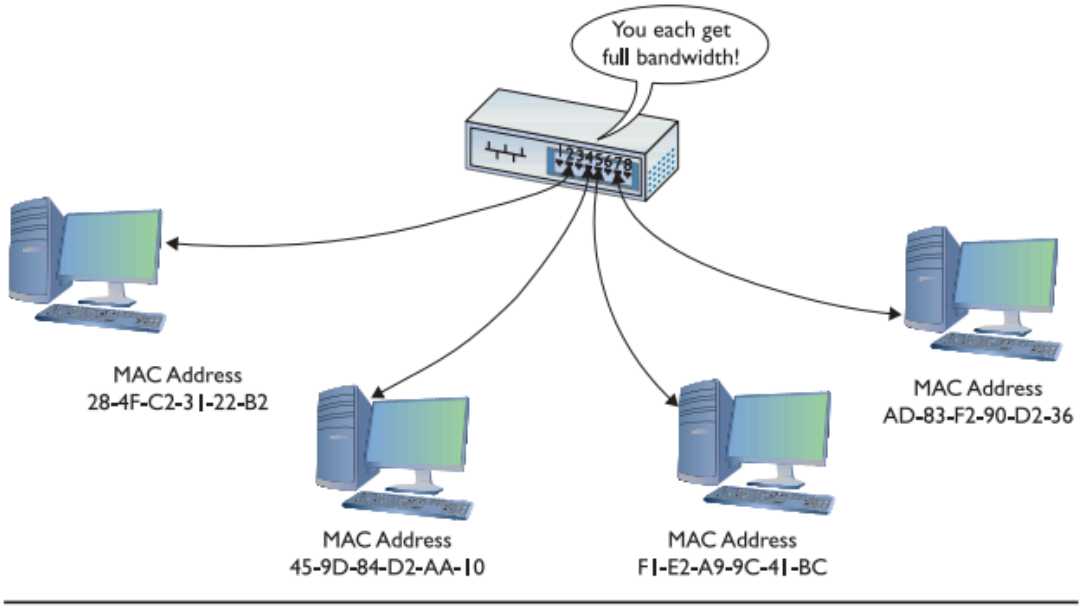


Figure 3-15 A switch making four separate connections

Each port on a switch is in its own collision domain, plus the switch can buffer incoming frames. That means that two nodes connected to the switch can send data at the same time and the switch will handle it without any collision.

NOTE Because a switch filters traffic on MAC addresses (and MAC addresses run at Layer 2 of the OSI seven-layer model), they are sometimes called *Layer 2 switches*.

Unicast messages always go only to the intended recipient when you use a switch and the switch knows the destination address. The switch sends all broadcast messages to all the ports (except the port on which the frame originated). You'll commonly hear a switched network called a *broadcast domain* to contrast it to the ancient hub-based networks with their collision domains.

Connecting Ethernet Segments

Sometimes, one switch is just enough. Once an organization uses every port on its existing switch, adding more nodes requires adding switches. Physical distance requirements also lead to the need for more switches. Even fault tolerance can motivate an organization to add more switches. If every node on the network connects to the same switch, that switch becomes a single point of failure—if it fails, everybody drops off the network. You can connect switches in two ways: via an uplink port or a crossover cable.

Uplink Ports

Uplink ports enable you to connect two switches using a straight-through cable. They're clearly marked on older switches, as shown in Figure 3-16. To connect two switches, insert one end of a cable in the uplink port and the other end of the cable in any one of the regular ports on the other switch.

Figure 3-16
Typical uplink port

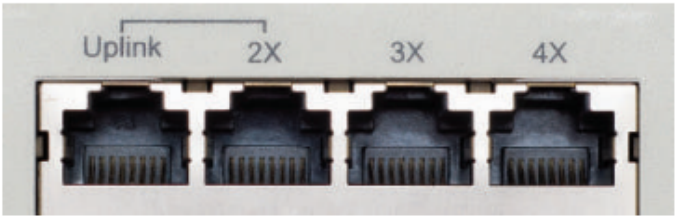


Figure 3-16 Typical uplink port

Modern switches do not have a dedicated uplink port, but instead auto-sense when another switch is plugged in. You can plug into any port.

EXAM TIP The technical term for an uplink port and the auto-sensing feature of ports in modern switches is *auto-medium-dependent interface crossover (MDI-X)*.

Crossover Cables

Switches can also connect to each other via special twisted pair cables called crossover cables. A *crossover cable* reverses the sending and receiving pairs on one end of the cable. One end of the cable is wired according to the TIA/EIA 568A standard, whereas the other end is wired according to the TIA/EIA 568B standard (Figure 3-17). With the sending and receiving pairs reversed, the switches can hear each other; hence the need for two standards for connecting RJ-45 jacks to UTP cables.

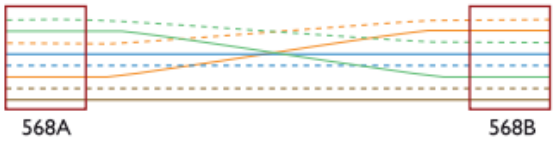


Figure 3-17 A crossover cable reverses the sending and receiving pairs.

A crossover cable connects to a regular port on each switch. Modern switches with auto-sensing ports don't require a crossover cable. In a pinch, you can use a crossover cable to connect two computers together using Ethernet NICs with no switch between them at all. This is handy for quickie connections, although not used much anymore because we mostly go wireless now.

EXAM TIP The CompTIA Network+ exam objectives list using a crossover cable in a troubleshooting scenario, presumably meaning you'd need to add an additional switch to a network to test ports. Modern switches auto-sense, so this isn't a thing anymore. If you're asked about interconnecting ancient switches (10BASE-T/100BASE-T), *crossover* might be the best answer.

Spanning Tree Protocol

Because you can connect switches together in any fashion, you can create redundant connections in a network. These are called *switching loops* or *bridge loops* (Figure 3-18).

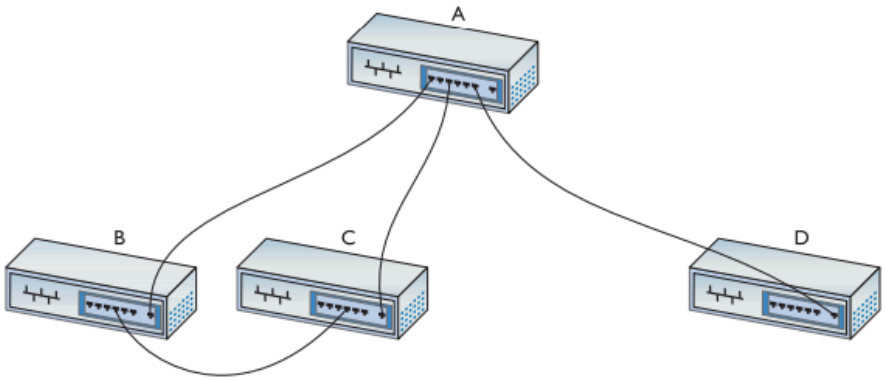


Figure 3-18 A switching loop

In the early days of switches, making a bridge loop in a network setup would bring the network crashing down. A frame could get caught in the loop, so to speak, and not reach its destination. The Ethernet standards body adopted the *Spanning Tree Protocol (STP)* to eliminate the problem of accidental switching loops. For decades, switches have had STP enabled by default,

and can detect potential loops before they happen. Using special STP frames known as *bridge protocol data units (BPDUs)*, switches communicate with other switches to prevent loops from happening in the first place.

Configuration BPDUs establish the topology, where one switch is elected as the root bridge and acts as the center of the STP universe. Each switch then uses the root bridge as a reference point to maintain a loop-free topology. There will be redundant links, for fault tolerance, that would ordinarily cause a switching loop, but certain ports will be placed in a “blocking” state and will not send or receive data frames. Ports in the blocking state will still hear the configuration BPDUs, which are sourced by the root bridge and forwarded downstream to the other switches every 2 seconds. If a link or device goes down, STP springs into action with another type of BPDU, called a *topology change notification (TCN) BPDU*, that enables the switches to rework themselves around the failed interface or device. The blocked ports, listening to the BPDUs, will realize they’re needed and eventually move to a forwarding state. Administrators can manually change STP settings for a switch. A switch port directly connected to a PC, for example, should never participate in STP, and could be configured with a setting called *PortFast* that enables the interface to come up right away, without the normal latency introduced by STP. Another reason to configure switch ports with PortFast is to prevent TCN BPDUs being sent out of that switch every time a PC is powered on and off, which has severe side effects, like causing all switches to flush their source address table, and relearn MAC addresses.

BPDU guard will move a port configured with PortFast into an errdisable state (i.e., error occurred, disabled) if a BPDU is received on that port. This requires an administrator to manually bring the port back up. Ports configured with PortFast should never receive a BPDU, and if they do, it could start a switching loop. Another mechanism, *root guard*, will move a port into a root inconsistent state if BPDUs coming from a certain direction indicate another switch is trying to become the root bridge. The root-inconsistent port will automatically return to its forwarding state once these BPDUs stop. This helps define locations where the root bridge should never be located.

NOTE The preceding terms used to describe functions within a switch apply specifically to Cisco switches. Cisco creates many of the network boxes (switches, routers, and more) that power a zillion networks, including much of the Internet. Other companies, notably Juniper, compete with Cisco and use different terms for the same actions.

The original Spanning Tree Protocol, introduced by IEEE as 802.1d, was replaced a long time ago (2001) by the Rapid Spanning Tree Protocol (RSTP), 802.1w. RSTP offers significantly faster convergence time following some kind of network change. STP could take up to 50 seconds to get back to a steady state, for example, whereas an RSTP network could return to convergence in 6 seconds.

Troubleshooting Switches

The simple switches described in this chapter generally function flawlessly for years without any need for a tech to do more than wipe dust off the top. Very occasionally you'll run into a switch that has problems. These problems often fall into two categories:

- Obvious physical damage
- Dead ports

Diagnosing any of these problems follows a similar pattern. First, you recognize that a switch might have problems because a device you've plugged in can't connect to the network. Second, you examine the switch for obvious damage. Third, you look for link lights. If they're not flashing, try a different port. Fourth, you look at your cables. If anything looks bent, broken, or stepped on, you should replace it. A bad cable or improper cable type can lead to problems that point to a "failed" switch when the true culprit is really the cable. Finally, you use the tried-and-true method of replacing the switch or the cable with a known-good device.

cable with a known-good device.

NOTE When we get to modern higher-end switches in Chapter 11, you'll need to follow other procedures to do proper diagnostic work. We'll get there soon enough!

NOTE When we get to modern higher-end switches in Chapter 11, you'll need to follow other procedures to do proper diagnostic work. We'll get there soon enough!

Chapter Review

Questions

1. Ethernet hubs took an incoming packet and _____ it out to the other connected ports.
    a. amplified
    b. repeated
    c. filtered
    d. distorted
2. What is at the beginning of the Ethernet frame?
    a. MAC address
    b. Length
    c. Preamble
    d. CRC
3. What type of bus did 10BASE-T use?
    a. Bus
    b. Ring
    c. Star bus

    d. Bus ring
4. What was the maximum distance that could separate a 10BASE-T node from its hub?
    a. 50 meters
    b. 100 meters
    c. 185 meters
    d. 200 meters
5. When used for Ethernet, unshielded twisted pair uses what type of connector?
    a. RG-58
    b. RJ-45
    c. RJ-11
    d. RS-232
6. What was the maximum number of nodes that could be connected to a 10BASE-T hub?
    a. 1024
    b. 500
    c. 100
    d. 185
7. Which of the following is not true of crossover cables?
    a. They are a type of twisted pair cabling.
    b. They reverse the sending and receiving wire pairs.
    c. They are used to connect switches.
    d. Both ends of a crossover cable are wired according to the TIA/EIA 568B standard.
8. Which of the following connectors were used by 10BASE-FL cable? (Select two.)
    a. SC
    b. RJ-45
    c. RJ-11
    d. ST
9. Which networking devices use the spanning tree protocol (STP)?
    a. Hubs
    b. Media converters
    c. UTP cables
    d. Switches
10. Which device directs packets based on MAC addresses?
    a. Router
    b. Hub
    c. Repeater
    d. Switch

Answers

1. B. Hubs were nothing more than multiport repeaters. 2. C. At the front of the Ethernet frame is the preamble. 3. C. 10BASE-T used a star-bus topology. 4. B. The maximum distance between a 10BASE-T node and its hub was 100 meters. 5. B. UTP cable uses an RJ-45 connector when used for Ethernet. RG-58 is the type of coaxial cable used with 10BASE-2.

RJ-11 is the standard four-wire connector used for regular phone lines. RS-232 is a standard for serial connectors. 6. A. A 10BASE-T hub could connect no more than 1024 nodes (computers). 7. D. One end of a crossover cable is wired according to the TIA/EIA 568B standard; the other is wired according to the TIA/EIA 568A standard. This is what crosses the wire pairs and enables two switches in early Ethernet to communicate. 8. A, D. 10BASE-FL used two types of fiber-optic connectors called SC and ST connectors. 9. D. The Spanning Tree Protocol is unique to switches. 10. D. A switch uses MAC addresses to direct traffic only to the appropriate recipient.

Chapter 4 - Ethernet Standards

The CompTIA Network+ exam expects you to know how to:

- 1.2 Explain the characteristics of network topologies and network types
- 1.3 Summarize the types of cables and connectors and explain which is the appropriate type for a solution

To achieve these goals, you must be able to:

- Describe the varieties of 100-bit Ethernet
- Discuss copper- and fiber-based Gigabit Ethernet
- Discover and describe Ethernet varieties beyond Gigabit

Within a few years of its introduction, 10BASE-T proved inadequate to meet the growing networking demand for speed. As with all things in the computing world, bandwidth is the key. Even with switching, the 10-Mbps speed of 10BASE-T, seemingly so fast when first developed, quickly found a market clamoring for even faster speeds. This chapter looks at the improvements in Ethernet since 10BASE-T. You'll read about the 100-megabit standards and the Gigabit Ethernet standards. The chapter finishes with a look at Ethernet standards that exceed Gigabit speeds.

Test-Specific - 100-Megabit Ethernet

The quest to break 10-Mbps network speeds in Ethernet started in the early 1990s. By then, 10BASE-T Ethernet had established itself as the most popular networking technology (although other standards, such as IBM's Token Ring, still had some market share). The goal was to create a new speed standard that made no changes to the Ethernet frames. By doing this, the 802.3 committee ensured that different speeds of Ethernet could interconnect, assuming you had something that could handle the speed differences and a media converter if the connections differed. This standardization ensures communication and scalability.

100BASE-T

When it came time to come up with a new standard to replace 10BASE-T, network hardware makers forwarded a large number of potential standards, all focused on the prize of leading the

new Ethernet standard. As a result, two twisted-pair Ethernet standards appeared: *100BASE-T4* and *100BASE-TX*. 100BASE-T4 used Cat 3 cable, whereas 100BASE-TX used Cat 5 and Cat 5e. By the late 1990s, 100BASE-TX became the dominant 100-megabit Ethernet standard. 100BASE-T4 disappeared from the market and today has been forgotten. As a result, we never say 100BASE-TX, simply choosing to use the term *100BASE-T*.

NOTE 100BASE-T was at one time called *Fast Ethernet*. The term still sticks to the 100-Mbps standards even though there are now much faster versions of Ethernet.

100BASE-T Summary

- **Speed** 100Mbps
- **Signal type** Baseband
- **Distance** 100 meters between the hub/switch and the node
- **Node limit** No more than 1024 nodes per hub/switch
- **Topology** Star-bus topology: physical star, logical bus
- **Cable type** Cat 5 or better UTP or STP cabling with RJ-45/8P8C connectors

EXAM TIP A *baseband* network means that only a single signal travels over the wires of the network at one time, occupying the lowest frequencies. Ethernet networks are baseband. Contrast this with *broadband*, where you can get multiple signals to flow over the same wire at the same time, modulating to higher frequencies. The latter is how cable television and cable Internet work.

Upgrading a 10BASE-T network to 100BASE-T was not a small process. First, you needed a Cat 5 cable or better. Second, you had to replace all 10BASE-T NICs with 100BASE-T NICs. Third, you had to replace the 10BASE-T hub or switch with a 100BASE-T hub or switch. Making this upgrade cost a lot in the early days of 100BASE-T, so people clamored for a way to make the upgrade a little easier and less expensive. This was accomplished via multispeed, auto-sensing NICs and hubs/switches.

Figure 4-1 shows a typical multispeed, auto-sensing 100BASE-T NIC from the late 1990s. When this NIC first connected to a network, it negotiated automatically with the hub or switch to determine the other device's highest speed. If they both did 100BASE-T, then you got 100BASE-T. If the hub or switch only did 10BASE-T, then the NIC did 10BASE-T. All of this happened automatically (Figure 4-2).

Figure 4-1
PCI 100BASE-T NIC

Figure 4-1 PC 100BASE-T NIC

Figure 4-2
Auto-negotiation in action

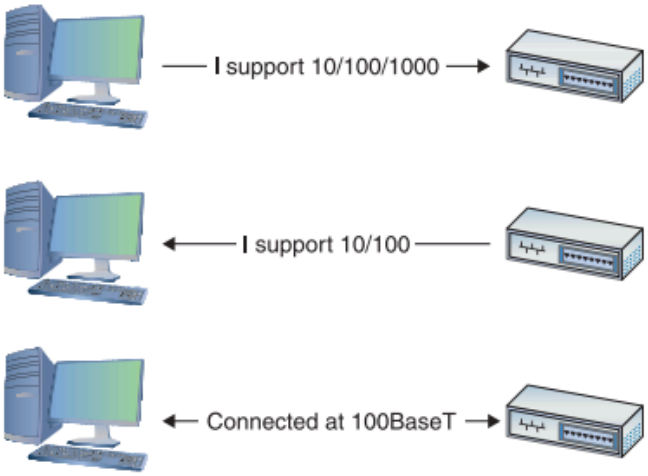


Figure 4-2 Auto-negotiation in action

Distinguishing a 10BASE-T NIC from a 100BASE-T NIC without close inspection was impossible. You had to look for something on the card to tell you its speed. Some NICs had extra link lights to show the speed (see Chapter 5 for the scoop on link lights). Of course, you could always simply install the card, as shown in Figure 4-3, and see what the operating system says it sees.

Figure 4-3
100BASE-T NIC in Windows 8.1

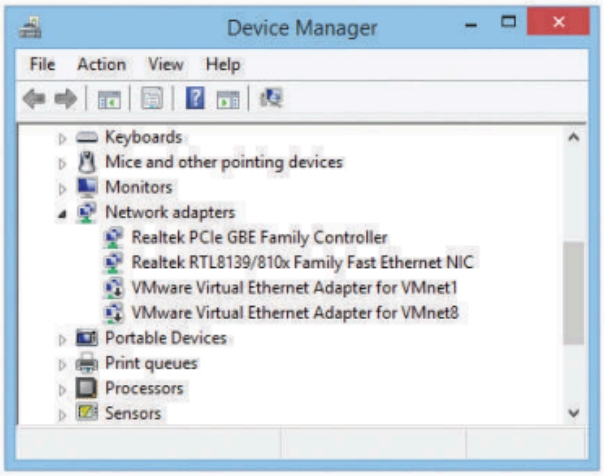


Figure 4-3 100BASE-T NIC in Windows 8.1

You'll also have trouble finding a true 10BASE-T or 100BASE-T NIC any longer because multispeed NICs have been around long enough to have replaced any single-speed NIC. All modern NICs are multispeed and auto-sensing.

CompTIA Network+ Deluxe Study Guide (Exam N10-004)

Deluxe Edition Bonus Content with Network+ Virtual Lab

Includes Real-World Scenarios, Written Labs, and Leading-Edge Exam Prep Software Featuring:

- Custom Test Engine with Six Practice Exams
- Exclusive Network+ Virtual Lab Network Simulator
- Hundreds of Sample Questions
- Electronic Flashcards
- Entire Book in PDF

CompTIA Network+ (™)

Most of us are running Transmission Control Protocol/Internet Protocol (TCP/IP) Version 4 on our networks these days so we absolutely need a way to test IP connectivity. But we also need be able to test and verify IPv6 networks. The reason for this is that even though Microsoft makes the majority of client platforms, a lot of these commands are really platform independent, and most of them can now use both IPv4 and IPv6. Even so, keep in mind that the Network+ exam focuses on the basic concepts of the function and use of the TCP/IP utilities that come with Windows. You can use several utilities to verify TCP/IP function on Windows workstations, and most of them are listed in the chapter objectives. But there are a few others that I'm going to discuss with you because they're really important for anyone working in the networking field to know about. Here's a list of them:

N Traceroute (tracert in other environments) ÛN ipconfig/winipcfg (ifconfig in Unix) ÛN ping ÛN arp ÛN nslookup (dig in Unix) ÛN Mtr ÛN route ÛN nbtstat ÛN netstat ÛN ftp ÛN Telnet

And by the way… it's very important that you don't just blow through the output that I've supplied for each command. Instead, pay serious attention to it, because to meet the Network+ objectives, you'll be required to correctly identify each command's output. So, let's cut right to the chase and take a look at some of these commands and their output. Oh, and do try and have fun with it!

For up-to-the-minute updates for this chapter, please see www.lammle.com or *www.sybex.com/go/comptianetwork+studyguide.*

Using *Traceroute*

For starters, let's pose these questions: "Where do all those packets really go when we send them over the Internet? And, how do all the packets actually get to their destinations?" Well, we can use the TCP/IP traceroute (tracert with Windows) command-line utility to help us answer both questions because its output will show us every router interface a TCP/IP packet passes

through on the way to its destination. Traceroute (trace for short), displays the path a packet takes to get to a remote device in all its glory by using something we call time to live (TTL), time-outs, and Internet Control Message Protocol (ICMP) error messages. And it's also a handy tool for troubleshooting an internetwork because we can use it to figure out which router along a path through that internetwork happens to be causing a network failure when a certain destination machine or network is, or suddenly becomes, unreachable. To use tracert, at a Windows command prompt, type tracert, a space, and the Domain Name Service (DNS) name or IP address of the host machine you want to find the route to. The tracert utility will respond with a list of all the DNS names and IP addresses of the routers that the packet is passing through on its way. Plus, tracert uses TTL to indicate the time it takes for each attempt. Following is the tracert output from my workstation in Boulder, Colorado to my Lammle.com server in Dallas, Texas:

C:\Users\tlammle> **tracert www.lammle.com**

Tracing route to lammle.com [206.123.114.186] over a maximum of 30 hops:

1 1 ms <1 ms <1 ms dslmodem.domain.actdsltmp [192.168.0.1] 2 53 ms 52 ms 52 ms hlrn-dsl-gw36-228.hlrn.qwest.net [207.225.112.228] 3 52 ms 53 ms 52 ms hlrn-agw1.inet.qwest.net [71.217.189.25] 4 75 ms 75 ms 74 ms dal-core-01.inet.qwest.net [67.14.2.53] 5 76 ms 76 ms 76 ms dap-brdr-01.inet.qwest.net [205.171.225.49] 6 76 ms 76 ms 76 ms 205.171.1.110 7 75 ms 76 ms 106 ms xe-0-0-0.er2.dfw2.us.above.net [64.125.26.206] 8 76 ms 76 ms 76 ms 209.249.122.74.available.above.net [209.249.122.74] 9 76 ms 76 ms 76 ms 65.99.248.250 10 76 ms 76 ms 76 ms pageuppro.pageuppro.com [206.123.114.186] Trace complete.

Okay, were you able to see that the packet bounces through several routers before arriving at its destination? Good! This utility is useful if you are having problems reaching a web server on the Internet nd you want to know if a wide area network (WAN) link is down, or if the server just isn't responding. What this means to you is that basically, wherever the trace stops is a great place to start troubleshooting. No worries here, though—the previous output shows that every router is up and responding. Lastly, notice in the output the "ms." This is the latency of each hop, meaning the delay. tracert or traceroute is a great troubleshooting tool to find out where your network bottlenecks are. If you use traceroute or tracert and receive an asterisk, this indicates that the attempt to reach that router took longer than the default time-out value. This is very good to know because it can mean that either the router is extremely busy or that a particular link is slow. Another reason for getting an asterisk could be that the administrator has disabled the ICMP protocol on the router that the packet is trying to hop through. Why would someone want to do that? For security reasons, that's why. It happens to be a typical strategic move done on the router(s) that interface to the ISP to conceal their actual location so bad guys can't hack into them and therefore, into your internetwork. It's a good idea, and I highly recommend doing it.

NOTE If you are running traceroute and see repeating addresses and TTL timeouts, you probably have a routing loop.

Using ipconfig and ifconfig

The utilities known as ipconfig (in Windows), and ifconfig (in Unix/Linux/Mac) will display the current configuration of TCP/IP on a given workstation—including the current IP address, DNS configuration, Windows Internet Naming Service (WINS) configuration, and default gateway. In the following sections, we will discuss how to use both.

Using the *ipconfig utility*

With the new Macs, Vista, and Windows Server 2008, you can see the IPv6 configuration because IPv6 is enabled by default. The output of the ipconfig command provides the basic routed protocol information on your machine. From a DOS prompt, type ipconfig, and you'll see something like this:

C:\Users\tlammle>**ipconfig**

Windows IP Configuration Ethernet adapter Local Area Connection: Connection-specific DNS Suffix . : domain.actdsltmp Link-local IPv6 Address . . . . . : fe80::2836:c43e:274b:f08c%11 IPv4 Address. . . . . : 192.16

Subnet Mask . . . . . : 255.255.255.0 Default Gateway . . . . . : 192.168.0.1 Wireless LAN adapter Wireless Network Connection: Connection-specific DNS Suffix . : qwest.net Link-local IPv6 Address . . . . . : fe80::20e7:7fb8:8a00:832b%10 IPv4 Address. . . . . : 10.0.1.198 Subnet Mask . . . . . : 255.255.255.0 Default Gateway . . . . . : fe80::21b:63ff:fef3:3694%10 10.0.1.1 Tunnel adapter Local Area Connection* 6: Media State . . . . . : Media disconnected Connection-specific DNS Suffix . : Tunnel adapter Local Area Connection* 7: Media State . . . . . : Media disconnected Connection-specific DNS Suffix . : [output cut for brevity]

Wow, there are a lot of options in this output compared to earlier versions of Windows! First, what's up with all these interfaces showing? I only have two – one Ethernet and one Wireless. ou can see that my Ethernet adapter shows up first, and it has an IP address, a mask, and a default gateway, plus an IPv6 address and a DNS suffix. The next configured interface is the wireless local area network (LAN) adapter, which has an IP address, a mask, a default gateway, an IPv6 address, and the IPv6 default gateway, as well. This IPv6 default gateway address is simply my router advertising that it runs IPv6 and "I am the way out of the local LAN!" The next adapters are disconnected because they are logical interfaces, and I'm not using them—my machine actually shows eight, but I cut the output because they provided no new information. They're automatically inserted because IPv6 is installed and running on my machine, and these adapters allow me to run IPv6 over an IPv4 only network. But just in case the ipconfig doesn't

provide enough information for you, try the ipconfig /all command—talk about details. Here's the beginning of that output:

```
C:\Users\tlammle>ipconfig /all
Windows IP Configuration
   Host Name . . . . . . . . . . . . : globalnet-todd
   Primary Dns Suffix  . . . . . . . : globalnet.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
```



```
   DNS Suffix Search List. . . . . . : globalnet.local
                                       domain.actdsltmp
                                       qwest.net
```

Ethernet adapter Local Area Connection: Connection-specific DNS Suffix . : domain.actdsltmp Description . . . . . : Intel(R) 82566MM Gigabit Network Connection Physical Address. . . . . : 00-1E-37-D0-E9-35 DHCP Enabled. . . . . : Yes Autoconfiguration Enabled . . . . : Yes Link-local IPv6 Address . . . . . : fe80::2836:c43e:274b:f08c%11(Preferred) IPv4 Address. . . . . : 192.168.0.6(Preferred) Subnet Mask . . . . . : 255.255.255.0 Lease Obtained. . . . . : Monday, October 20, 2008 9:08:36 AM Lease Expires . . . . . : Tuesday, October 21, 2008 9:08:39 AM Default Gateway . . . . . : 192.168.0.1 DHCP Server . . . . . : 192.168.0.1 DNS Servers . . . . . : 192.168.0.1 205.171.3.65 NetBIOS over Tcpip. . . . . : Enabled Wireless LAN adapter Wireless Network Connection: Connection-specific DNS Suffix . : qwest.net Description . . . . . : Intel(R) Wireless WiFi Link 4965AGN Physical Address. . . . . : 00-1F-3B-3F-4A-D9 DHCP Enabled. . . . . : Yes Autoconfiguration Enabled . . . . : Yes Link-local IPv6 Address . . . . . : fe80::20e7:7fb8:8a00:832b%10(Preferred) IPv4 Address. . . . . : 10.0.1.198(Preferred) Subnet Mask . . . . . : 255.255.255.0 Lease Obtained. . . . . : Monday, October 20, 2008 10:43:53 AM Lease Expires . . . . . : Monday, October 20, 2008 2:43:53 PM Default Gateway . . . . . : fe80::21b:63ff:fef3:3694%10 10.0.1.1 DHCP Server . . . . . : 10.0.1.1 DNS Servers . . . . . : 10.0.1.1 NetBIOS over Tcpip. . . . . : E

Tunnel adapter Local Area Connection* 6:

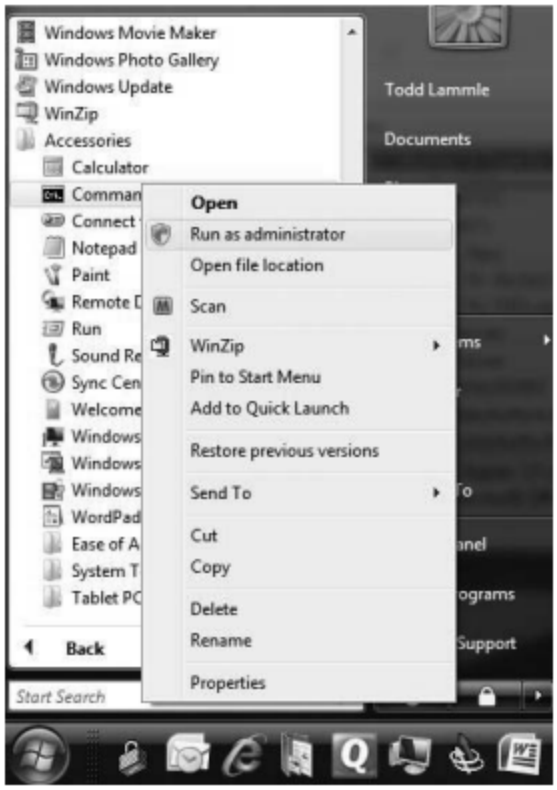
Media State. . . . . : Media disconnected
Connection-specific DNS Suffix. : isatap.globenet.local
Physical Address. . . . . :00-00-00-00-00-00-00-E0
DHCP Enabled. . . . . :No
Autoconfiguration Enabled. . . . . :Yes
[output cut]

As you can see, it's more of the same – a whole lot more. The most important thing I want you to notice is that I've received the hardware information about each interface, including the Media Access Control (MAC) address. Also significant is that I can see the Dynamic Host Configuration Protocol (DHCP) lease times and DNS addresses now. But why stop here? There are two more valuable options you need to use with the ipconfig command. They are /release and /renew. When you change networks, you need to get the IP address of that subnet and/or virtual LAN (VLAN). Vista works most of the time without doing anything, but sometimes I do have to renew the IP configuration when changing networks. But that's easy—just type ipconfig /renew from a command prompt, and if you're connected to DHCP server that's available, you'll then magically receive an IP address. Now, if it still doesn't work, you'll need to release and renew your TCP/IP settings. To release your current DHCP TCP/IP information, you must elevate your command prompt, or you'll get this warning:

C:\Users\tlammle> **ipconfig /release**
The requested operation requires elevation.
C:\Users\tlammle>

Should this happen to you, go to Start All Programs Accessories Command Prompt, right-click, and choose Run As Administrator. (Of course, you'll have to enter your name and password to do this if you are using Vista. But we love Vista, right? Okay, maybe not always.) Anyway, Figure 17.1 shows how I did this. Once your Command Prompt has been duly elevated, you can use the ipconfig /release command and then the ipconfig /renew command to get new TCP/IP information for your host.

Figure 17.1 Elevating your command prompt



Using the *ifconfig* utility

There is a utility in Linux/Unix/Mac that will give you information similar to what ipconfig shows. It's called ifconfig (short for "interface configuration"). Although ipconfig and ifconfig show similar information, there are major differences between these two utilities. The ipconfig utility is mainly used to view the TCP/IP configuration for a computer. You can use ifconfig to do the same thing, but ifconfig can also be used to configure a protocol or a particular network interface. The general syntax of the ifconfig command is as follows:

Ifconfig *interface [address* [parameters]]

The *interface* parameter equals the Unix name of the interface, such as eth0. If the optional address parameter is specified, the ifconfig command sets the IP address for the interface to the address you've specified. When the ifconfig command is used by itself with no parameters, all configured interfaces will be reported on. But if only the interface name is specified, you'll get output that looks like this:

 # ifconfig eth0 eth0 Link encap 10Mbps Ethernet HWaddr 00:00:C0:90:B3:42 inetaddr 172.16.0.2 Bcast 172.16.0.255 Mask 255.255.255.0 UP BROADCAST RUNNING MTU 1500 Metric 0

RX packets 3136 errors 217 dropped 7 overrun 26 TX packets 1752 errors 25 dropped 0 overrun 0 Looking at this, we can see that the eth0 interface is a 10Mbps Ethernet interface. The interface's MAC and IP address information is displayed in this output as well. And, although not shown in the output, the ifconfig tool can show you the DNS information configured on the host.

Using the *ping* utility

Ping is the most basic TCP/IP utility, and it's included with most TCP/IP stacks for most platforms. Windows, again, is no exception. In most cases, ping is a command-line utility, although there are many GUI implementations available. You use the ping utility for two primary purposes:

- To find out if a host is responding
- To find out whether you can reach a host

Here's the syntax:

Ping *hostname or IP address*

Ping hostname or IP address If you ping any station that has an IP address, the ICMP that's part of that particular host's TCP/IP stack will respond to the request. This ICMP test and response looks something like this: ping 204.153.163.2 Pinging 204.153.163.2 with 32 bytes of data: Reply from 204.153.163.2: bytes=32 time<10ms TTL=128 Reply from 204.153.163.2: bytes=32 time=1ms TTL=128 Reply from 204.153.163.2: bytes=32 time<10ms TTL=128 Reply from 204.153.163.2: bytes=32 time<10ms TTL=128 Because I've received a reply from the destination station (204.153.163.2, in this case), I know that I can reach the host and that it's responding to basic IP requests. Don't forget that you can use name resolution and ping to a name such as ping www.sybex.com, and as long as that name can be resolved, you're golden. Most versions of ping work the same way, but there are some switches you can use to specify certain information like the number of packets to send, how big a packet to send, and so on. And if you're running the Windows command-line version of ping, just use the /? or –? switch to display a list of the available options like this:

C:\Users\tlammle> ***ping /?***

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name

The command will also output a table showing what each of the options does, presented here in Table 17-1.

TABLE 17.1 Options for *ping Switches*

TABLE 17.1 Options for *ping Switches*

Option	Description
`-t`	Pings the specified host until stopped. To see statistics and continue, press Ctrl+Break; to stop, press Ctrl+C.
`-a`	Resolves addresses to hostnames.
`-n` *count*	Specifies the number of echo requests to send.
`-l` *size*	Sends the buffer size.
`-f`	Sets the Don't Fragment flag in the packet (IPv4-only).
`-i` *TTL*	Specifies the time to live.
`-v` *TOS*	Specifies the type of service (IPv4-only).
`-r` *count*	Records the route for count hops (IPv4-only).
`-s` *count*	Specifies the timestamp for count hops (IPv4-only).
`-j` *host-list*	Uses a loose source route along the host-list (IPv4-only).
`-k` *host-list*	Uses a strict source route along host-list (IPv4-only).
`-w` *timeout*	Specifies the timeout in milliseconds to wait for each reply.
`-R`	Uses the routing header to test the reverse route also (IPv6-only).
`-S` *srcaddr*	Specifies the source address to use.
`-4`	Forces using IPv4.
`-6`	Forces using IPv6.

TIP You can ping your local TCP/IP interface by typing ping 127.0.0.1 or ping localhost. Understand that both addresses represent the local interface.

As you can see, there's a plethora of options you can use with the ping command from a Windows DOS prompt. But I really want you to focus on a few from the previous output. (I'm only going to go over a few of them, but you can get on your host machine and play with all the options.) The -a switch is very cool because if you have name resolution (such as a DNS server), you can see the name of the destination host even if you only know its IP address. The -n switch sets the number of echo requests to send, where four is the default, and the -w switch allows you to adjust the time-out in milliseconds. The default ping timeout is 1 second (1000ms). The -6 is also nice if you want to ping an IPv6 host. By the way, unless you really love typing 128-bit addresses, this is a wonderful example of how important name resolution is. And then

there's -t, which keeps the ping running. Here's an example of a ping to an IPv6 address: C:\Users\tlammle>ping -6 fe80::1063:16af:3f57:fff9 Pinging fe80::1063:16af:3f57:fff9 from fe80::1063:16af:3f57:fff9%25 with 32 bytes of data: Reply from fe80::1063:16af:3f57:fff9: time<1ms Reply from fe80::1063:16af:3f57:fff9: time<1ms Reply from fe80::1063:16af:3f57:fff9: time<1ms Reply from fe80::1063:16af:3f57:fff9: time<1ms Ping statistics for fe80::1063:16af:3f57:fff9: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms C:\Users\tlammle> And if I want to have a continuous ping, I just use that -t option like this: C:\Users\tlammle>ping -t 192.168.0.1

Pinging from 192.168.0.1 with 32 bytes of data:

Reply from 192.168.0.1: bytes=32 time=7ms TTL=255
Reply from 192.168.0.1: bytes=32 time=1ms TTL=255
Reply from 192.168.0.1: bytes=32 time=1ms TTL=255
Reply from 192.168.0.1: bytes=32 time=1ms TTL=255
Reply from 192.168.0.1: bytes=32 time=1ms TTL=255
Reply from 192.168.0.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.0.1:
Packets: Sent = 6, Received = 6, Lost = 0 (0% loss),
Approximate round-trip in milliseconds:

Minimum = 1s, Maximum = 7ms, Average = 2ms
Control - C
^C
C:\Users\tlammle>

This ping will just keep going and going like the Energizer Bunny until you press Ctrl+C. And by the way, it's an awesome tool for troubleshooting links.

Using the Address Resolution Protocol (ARP)

The *Address Resolution Protocol (ARP)* is part of the TCP/IP protocol stack. It's used to translate TCP/IP addresses to MAC addresses using broadcasts. When a machine running TCP/IP wants to know which machine on an Ethernet network is using a certain IP address, it will send an ARP broadcast that says, in effect, "Hey… exactly who is IP address xxx.xxx.xxx.xxx?" The machine that owns the specific address will respond with its own MAC address, supplying the answer. The machine that made the inquiry will respond by adding the newly gained information to its own ARP table. In addition to the normal usage, the ARP designation refers to a utility in Windows that you can use to manipulate and view the local workstation's ARP table.

The Windows ARP Table

The *ARP table* in Windows includes a list of TCP/IP addresses and their associated physical (MAC) addresses. This table is cached in memory so that Windows doesn't have to perform ARP lookups for frequently accessed TCP/IP addresses like those of servers and default gateways. Each entry contains an IP address and a MAC address, plus a value for TTL that determines how long each entry will remain in the ARP table. Remember that the ARP table contains two kinds of entries:

- Static
- Dynamic

Dynamic ARP table entries are created whenever the Windows TCP/IP stack performs an ARP lookup but the MAC address isn't found in the ARP table. When the MAC address of the requested IP address is finally found, or resolved, that information is then added into the ARP table as a dynamic entry. Whenever a request to send a packet to the host is sent to the Data Link layer, the ARP cache is checked first before an ARP broadcast is sent out. Remember, the ARP request is broadcast on the local segment—it does not go through a router.

NOTE The ARP table is cleared of dynamic entries whose TTL has expired to ensure that the entries are current.

Static ARP table entries serve the same function as dynamic entries but are made manually using the arp utility.

Using the *arp* Utility

Okay—you now know that ARP is a protocol included the TCP/IP suite. You also understand that ARP is used by IP to determine the MAC address of a device that exists on the same subnet as the requesting device. When a TCP/IP device needs to forward a packet to a device on the local subnet, it first looks in its own table, called an ARP cache, for an association between the known IP address of the destination device on the local subnet and that same device's MAC address. The cache is called that because the contents are periodically weeded out. If no association that includes the destination IP address can be found, the device will then send out an ARP broadcast that includes its own MAC and IP information as well as the IP address of the target device and a blank MAC address field. Filling in that blank is the object of the whole operation—it's the unknown value that the source device is requesting to be returned to it in the form of an ARP reply. Windows includes a utility called arp that allows us to check out the operating system's ARP cache. To view this, from a Windows DOS prompt, use the arp command like this:

C:\Users\tlammle> **arp**

Displays and modifies the IP-to-Physical address translation tables used by address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

Table 17.1 describes the various options that you can use with the arp command.

TABLE 17.1 arp Option Descriptors

Table 17.1 describes the various options that you can use with the arp command.

TABLE 17.2 arp Option Descriptions

Option	Description
-a	Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.

TABLE 17.2 arp Option Descriptions *(continued)*

Option	Description
-a	Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.
-g	Same as -a.
-v	Displays current ARP entries in verbose mode. All invalid entries and entries on the loop-back interface will be shown.
inet_addr	Specifies an Internet address.
-N	Displays the ARP entries for the network interface specified by if_addr.
-d	Deletes the host specified by inet_addr. inet_addr may be wildcarded with * to delete all hosts.
-s	Adds the host, and associates the Internet address inet_addr with the physical address eth_addr. The physical address is given as six hexa-decimal bytes separated by hyphens. The entry is permanent.
eth_addr	Specifies a physical address.
if_addr	If present, specifies the Internet address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

Sheesh, Looking at that output really makes me wish we were all just running IPv6, because as you already should know, IPv6 doesn't need ARP as well as many other annoying features and protocols required when running IPv4. Of note, the Windows arp utility is primarily useful for resolving duplicate IP addresses. For example, let's say your workstation receives its IP address from a DHCP server, but it accidentally receives the same address as some other workstation gets. And so, when you try to ping it, you get no response. Your workstation is basically confused—it's trying to determine the MAC address, and it can't because two machines are reporting that they have the same IP address. To solve this little snag, you can use the arp utility to view your local ARP table and see which TCP/IP address is resolved to which MAC address. To display the entire current ARP table, use the arp command with the –a switch, like so:

C:\Users\tlammle> ***arp -a***

Interface: 192.168.0.6 --- 0xb
Internet Address Physical Address Type
192.168.0.1 00-15-05-06-31-b0 dynamic
192.168.0.255 ff-ff-ff-ff-ff-ff static

224.0.0.22 01-00-5e-00-00-16 static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
255.255.255.255 ff-ff-ff-ff-ff-ff static

Interface: 10.100.10.54 --- 0x10
Internet Address Physical Address Type
10.100.10.1 00-15-05-06-31-b0 dynamic
10.100.10.255 ff-ff-ff-ff-ff-ff static
224.0.0.22 01-00-5e-00-00-16 static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static

TIP By the way, the -g switch will produce the same result.

Now, from this output, you can tell which MAC address is assigned to which IP address. Then, for static assignments, you can tell which workstation has a specific IP address and if it's indeed supposed to have that address by examining your network documentation—you do have that record, right? For DHCP-assigned addresses, you can begin to uncover problems stemming from multiple DHCP scopes or servers doling out identical addresses and other common configuration issues. And remember that under normal circumstances, you shouldn't see IP addresses in the ARP table for a given interface that aren't members of the same IP subnet as the interface.

NOTE If the machine has more than one network card (as may happen in Windows servers and on laptops with both Ethernet and wireless cards), each interface will be listed separately.

It's good to know that in addition to displaying the ARP table, you can use the arp utility to manipulate the table itself. To add static entries to the ARP table, you use the arp command with the –s switch. These static entries will stay in the ARP table until the machine is rebooted. A static entry essentially hard-wires a specific IP address to a specific MAC address so that when a packet needs to be sent to that IP address, it will automatically be sent to that MAC address. Here's the syntax:

Arp -s [IP Address][MAC Address]

Simply replace the [IP Address] and [MAC Address] sections with the appropriate entries, like so: arp –s 204.153.163.5 00–a0–c0–ab–c3–11 Now, take a look at your new ARP table by using the arp –a command. You should see something like this: Internet Address Physical Address Type 204.153.163.5 00–a0–c0–ab–c3–11 static Finally, if you want to delete entries from the ARP table, you can either wait until the dynamic entries time out or use the –d switch with the IP address of the static entry you'd like to delete, *like this: arp –d 204.153.163.5 Doing so effectively deletes the entry from the ARP table in memory.*

NOTE The arp utility doesn't confirm successful additions or deletions (use arp -a or arp -g for that), but it will give you an error message if you use incorrect syntax.

Using the nslookup utility

Whenever you're configuring a server or a workstation to connect to the Internet, you've got to start by configuring DNS if you want name resolution to happen (that is, if you want to be able to type www.sybex.com instead of an IP address). When configuring DNS, it's a very good thing to be able to test what IP address DNS is returning to ensure that it's working properly. The nslookup utility allows you to query a name server and quickly find out which name resolves to which IP address.

NOTE The Unix *dig* (short for domain information *groper*) utility does the same exact thing as nslookup. It's primarily a command-line utility that allows you to perform a single DNS lookup for a specific entity, but it can also be employed in batch mode for a series of lookups. Detailed information on this command is beyond the scope of this study guide, but you can find more information on the Web by searching for "unix dig."

The nslookup utility comes with Windows NT and later, as well as most versions of Unix and Linux, but not with Windows 95/98. At the command prompt, you can start the nslookup utility by typing in **nslookup** and pressing Enter. When you're inside this utility, the command prompt will change from something similar to C:\> sign to a shorter > sign. It will also display the name and IP address of the default DNS server you will be querying (you can change it, if necessary). Now, you can start using nslookup. The following output gives you a sample of the display after the nslookup command has been entered at the C:\> prompt.

```
C:\Users\tlammle>nslookup
Default Server: gnt-corpdc1.globalnet.local Address: 10.100.36.12

>
```

The primary job of nslookup is to tell the many different features of a domain name, the names of the servers that serve them, and how they're configured. To get that, just type in a domain name at the > prompt, and the nslookup utility will then return this information:

```
> lammle.com
Server: dslmodem.domain.actdsltmp
Address: 192.168.0.1

Non-authoritative answer:
Name: lammle.com
Address: 206.123.114.186
```

What this tells you is that the server that returned the information is not responsible (authoritative) for the zone information of the domain for which you requested an address, and that the name server for the domain lammle.com is located at the IP address 206.123.114.186. You can also ask nslookup for other information by setting a different option within nslookup. Just type ***set option*** at the > prompt and replace option with the actual option you want to use, for example, >set type=mx to determine the IP address of your email server. If you can't decide which one you want, use the question mark (?) at the greater than sign (>) to see all available options.

If you type in nslookup and receive this reply:

NS request timed out.
    Timeout was 2 seconds.

***Can't find server name for address 206.123.114.186: Timed out Default Server: UnKnown Address: fec0:0:0:ffff::1

Then you know that your DNS servers are not answering. You need to get over to the DNS server stat!

Resolving Names with the Hosts Table

The Hosts table is really a lot like DNS, except its entries are static for each and every host and server. Within the Hosts table, you'll find a collection of host names that devices reference for name-resolution purposes. And even though it works in both IP and IPv6 environments, it's unlikely you will use it these days, because the Hosts table is a way-ancient relic left over from old Unix machines. But just because it's museum quality doesn't mean you won't run into it now and then, which is the main reason I'm talking to you about it. You can find the Hosts table in C:\Windows\System32\drivers\etc. Just double-click the file, and then choose to open the file in Notepad or other text editor. Here's the default information—it's really nothing more than an explanation of how to use it and the local hosts for both IP and IPv6:

```
# Copyright (c) 1993-2006 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

127.0.0.1       localhost
::1             localhost
```

NOTE Any information entered to the right of a pound sign (#) in a Hosts file is ignored, so that you can use this space for comments.

Because it's a plain ASCII text file, you add the IP address under the local hosts and then the name to which you want to resolve the IP address. It's a pretty simple configuration, and again, one I don't recommend using because you have to type in the names of every host on every machine in your network. DNS is definitely the name resolution of choice for networks today.

NOTE Do not get the hosts table confused with the hostname command. The hostname command doesn't do much but display the name of your host, as shown:

C:\Users\tlammle> ***hostname /?***

Prints the name of the current host. Hostname

C:\Users\tlammle> ***hostname***

Globalnet-todd

Using the Mtr Command

Mtr or My traceroute is a computer program that combines the functions of the traceroute and ping utilities in a single network diagnostic tool. It also adds round-trip time and packet loss to the output—very cool. Mtr probes routers on the route path by limiting the number of hops individual packets are allowed to traverse and listening to news of their termination. It will regularly repeat this process (usually once per second) and keep track of the response times of the hops along the path. Mtr is great if you have Linux or Unix, but by default, it's not installed on Windows devices. Third-party applications of Mtr are available to install on Windows, but Microsoft did respond with its own version of Mtr—it's called pathping and provides the same functions as Mtr. Here's a look at the output and the options:

C:\Users\tlammle> **pathping**

Usage: pathping [-g host-list] [-h maximum_hops] [-i address] [-n] [-p period] [-q num_queries] [-w timeout] [-4] [-6] target_name

Table 17.3 lists the options of the Windows pathping command.

TABLE 17.3 *pathping* Options and Descriptions

Option	Description
`-g host-list`	Uses a loose source route along the *host-list*.
`-h maximum_hops`	Specifies the maximum number of hops to search for the target.

TABLE 17.3 *pathping* Options and Descriptions *(continued)*

Option	Description
`-i address`	Uses the specified source address.
`-n`	Does not resolve addresses to hostnames.
`-p period`	Waits *period* milliseconds between pings.
`-q num_queries`	Specifies the number of queries per hop.
`-w timeout`	Waits *timeout* milliseconds for each reply.
`-4`	Forces using IPv4.
`-6`	Forces using IPv6.

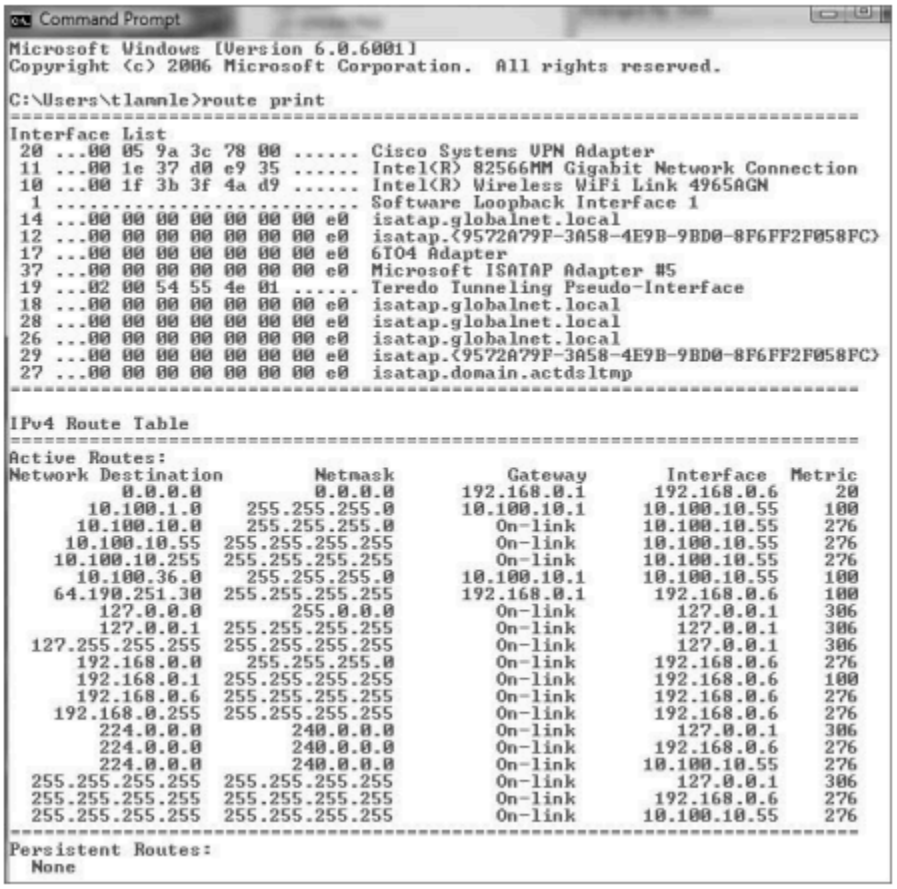
NOTE The *Mtr* utility is basically the same as traceroute and ping, but it does give you some additional output that can help you troubleshoot your network.

Using the Route Command

I went over static routing in Chapter 9, “Introduction to IP Routing,” so you know that Windows devices like routers perform routing. Most of the time, it’s a good idea to leave Windows alone, but it’s still good to know how to add and delete routes on your Windows machines. Probably the biggest reason for manipulating the routing table on a Windows server is to create a firewall. For instance, let’s say we’re running an application layer firewall on a Windows server located between the demilitarized zone (DMZ) and the internal network. This scenario would mean the routing that’s happening on the server or hosts located in the DMZ wouldn’t be able to reach the internal network’s hosts and vice versa. To circumvent this problem, we would need to employ

both static and default routing, because Windows Vista and Server 2008 don't support routing protocols—running routing protocols on hosts and servers wouldn't be a good solution for today's networks, and Microsoft knows that. To view the routing table on a Windows device, use the route print command, as shown in Figure 17.2.

Figure 17.2 route print output



In this output, you can see that each of the routes was added automatically when the system booted up. (This is all based on the configuration of your IP stack.) To see all the options available with the route command, type the route command and then press Enter. To add a route to your routing table, use the following syntax:

route [-f] [-p] [Command [Destination] [mask Netmask] [Gateway] [metric Metric]] [if Interface]]

Using the *route* Command Options

Let's start with the switches you can use: -f Using this command with any of the options like add, change, or delete will clear the routing table of all entries that aren't host routes (routes with the subnet mask 255.255.255.255), the loopback network route(s) (routes with a

destination of 127.0.0.0 and the subnet mask 255.0.0.0), and any multicast routes (those with a destination of 224.0.0.0 and the subnet mask 240.0.0.0). -p If you use this with the add command, the individual route will be added to the registry and then used to initialize the IP routing table whenever TCP/IP is started. Important to remember is that by default, the routes you've statically added won't remain in the routing table the next time TCP/IP boots. And if you use -p with the print command, you'll get shown a list of the persistent routes that are stored in the registry location of HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes. Now, let's take a look at how and when you would use the route command. Table 17.4 shows the command options available and what they do when using the route command with them.

Table 17.4 route Command options

TABLE 17.4 route Command Options

Command	Purpose
add	Adds a route
change	Modifies an existing route
delete	Deletes a route(s)
print	Prints a route(s)

Here's a description of some other tasks you can accomplish via the rest of the command's options:

Destination This will give you the network destination of a given route. If the host bits of the network address are set to 0, it will be depicted with the destination's IP network address, an IP address for a specific host route, or the default route of 0.0.0.0.

Mask netmask This will provide you with the *netmask*—often referred to as the *subnet mask*—that's associated with the destination network. The default destination subnet mask is 0.0.0.0, and typically you'll see 255.255.255.255 representing a host route. It's really important to remember that the destination address can't be more specific than its corresponding subnet mask. What I'm saying is that there absolutely can't be a bit set to 1 in the destination address if the equivalent bit in the subnet mask is a 0.

Gateway The gateway also depends on the network address and subnet mask, but it's even more specific and delimits what's called the *next-hop address.* For routes located on a local subnet, the gateway address maps directly to a particular interface. If the destination is on a remote network, the gateway IP address will direct packets to the neighboring router.

***Metric* metric**

Metric refers to the cost of a given route from the sending to the receiving device, and it's a value between 1 and 9999. Devices use this value to choose the best, or most efficient, routes among those in its routing table—the route with the lowest value wins. This decision can also include factors like the number of hops; the speed, reliability, and available bandwidth of the path being considered; plus the various administrative aspects associated with it.

If interface This tool depends on information from the gateway address and determines the interface index for the specific interface that needs to receive the data. You can get a list of interfaces along with their relevant interface indexes by typing the route print command.

/? Using this will allow you to view help at the command prompt.

Some Examples of the *Route* Command

Even though the finer points of the route command demand that you use caution when deploying some of them, I'll still list the basics of the route command because it can be really useful. I highly recommend that you spend some time practicing them on a nonproduction server, though—especially at first.

- To display the entire IP routing table, type **route print**.
- To add a default route with the default gateway address 192.168.10.1, type **route add 0.0.0.0 mask 0.0.0.0 192.168.10.1.**
- To add a route to the destination 10.1.1.0 with the subnet mask 255.255.255.0 and the next-hop address 10.2.2.2, type **route add 10.1.1..0 mask 255.255.255.0 10.2.2.2.**
- If you want to, let's say, add a persistent route to the destination 10.100.0.0 with the subnet mask 255.255.0.0 and the next-hop address 10.2.0.1, type **route -p add 10.100.0.0 mask 255.255.0.0 10.2.0.1.**If you want to delete the route to the destination 10.100.0.0 with the subnet mask 255.255.0.0, enter **route delete 10.100.0.0 mask 255.255.0.0.**

And finally, if you want to change the next-hop address of a route with the destination 10.100.0.0 and the subnet mask 255.255.0.0 from 10.2.0.1 to 10.7.0.5, type **route change 10.100.0.0 mask 255.255.0.0 10.7.0.5.**

Using the *nbstat* utility

Microsoft Windows uses an interface called Network Basic Input/Output System (NetBIOS), which relates names with workstations and is an upper-layer interface that requires a transport protocol—usually, TCP/IP. But IPv6 can be used as well. Deploying the nbtstat utility will achieve these three important things:

- Track NetBIOS over TCP/IP statistics
- Show the results of incoming and outgoing NetBIOS over TCP/IP connections

- Resolve NetBIOS names

Understand that because NetBIOS name resolution is primarily a Windows network utility, the nbtstat command is available only in Windows-based operating systems. To display a basic description of nbtstat and its associated options, type **nbstat** at the command line. Then, use these options to get a display of information about NetBIOS over TCP/IP hosts. Here are some of the tools, or switches, you can use:

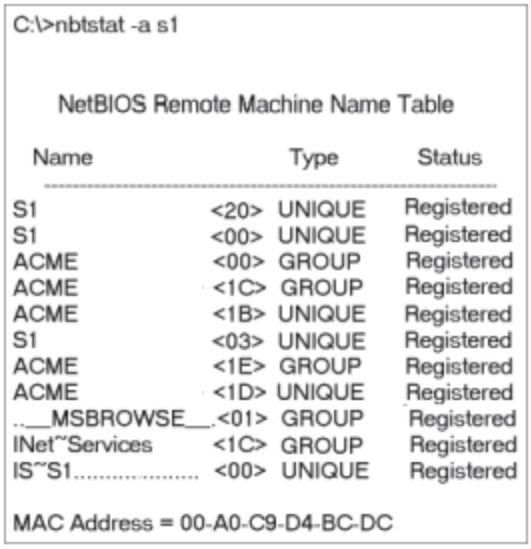
-a	-A
-c	-n
-r	-R
-S	-s

All nbtstat switches are case sensitive. Generally speaking, lowercase switches deal with NetBIOS names of hosts, and the uppercase ones deal with the TCP/IP addresses of hosts.

The *-a Switch*

Making use of the–a switch will get you a remote machine's NetBIOS name table consisting of a list of every NetBIOS name the machine you've deployed the switch from knows of. The –a switch produced the output from server S1 shown in Figure 17.3. So, using this switch arranges the NetBIOS name-table information in table form with output in four columns. The Name column displays the NetBIOS name entry for the remote host machine.

Figure 17.3 Sample output of the nbstat -a command



```
C:\>nbtstat -a s1

        NetBIOS Remote Machine Name Table

   Name                   Type         Status
   ---------------------------------------------------------
S1                <20>  UNIQUE      Registered
S1                <00>  UNIQUE      Registered
ACME              <00>  GROUP       Registered
ACME              <1C>  GROUP       Registered
ACME              <1B>  UNIQUE      Registered
S1                <03>  UNIQUE      Registered
ACME              <1E>  GROUP       Registered
ACME              <1D>  UNIQUE      Registered
..__MSBROWSE__.<01>  GROUP       Registered
INet~Services     <1C>  GROUP       Registered
IS~S1.....        <00>  UNIQUE      Registered

MAC Address = 00-A0-C9-D4-BC-DC
```

The next column gives you a unique two-digit hexadecimal identifier for the NetBIOS name. This identifier represents the last byte of the NetBIOS name depicted in the Name column, and it's important because the same name could actually be used several times for the same machine. Plus, it identifies the specific service on the particular host that the name is referencing. Tables 17.5 and 17.6 list the hexadecimal identifiers for unique and group host names.

TABLE 17.5 Last-Byte Identifiers for Unique Names

Hex ID	Description
00	General name for the computer.
03	Messenger service ID used to send messages between a WINS server and a workstation. This is the ID registered with a WINS server.
06	Remote Access Server (RAS) server service ID.
20	File-serving service ID.
21	RAS client.
53	DNS.
123	Network Time Protocol (NTP).
1B	Domain master browser ID. A NetBIOS name with this ID indicates the domain master browser.
1F	Network Dynamic Data Exchange (NetDDE) service ID.
BE	Network monitor agent ID.
BF	Network monitor utility ID.

Table 17-6 Last-Byte Identifiers for Group Names

Hex ID	Description
01	Master browser for a domain to other master browsers.
20	Internet group name ID. This ID is registered with the WINS server to indicate which computers are used for administrative purposes.

TABLE 17.6 Last-Byte Identifiers for Group Names *(continued)*

Hex ID	Description
1C	Domain group name ID.
1D	Master browser name.
1E	Normal group name.

The Type column refers to (surprise) the type of NetBIOS name being referenced. Unique NetBIOS names refer to individual hosts, and Group names refer to the names of logical groupings of workstations—either domains or workgroups. The Status column gives you information about the status of host's NetBIOS even if it hasn't been registered with the rest of the network.

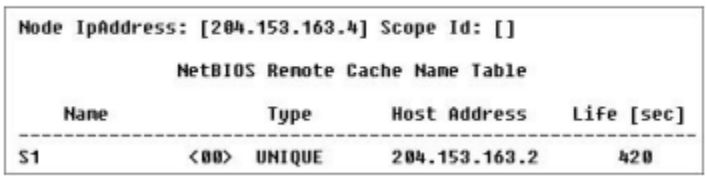
The *-A Switch*

The –A switch works just like the –a switch and will give you the same output, but the syntax of the command is different. Obviously, you use an uppercase A instead of a lowercase one, and you also have to include the host's IP address instead of its NetBIOS name. To use it, type nbtstat followed by –A and finally the IP address of the specific host whose NetBIOS table you want to check out: nbtstat –A 199.153.163.2

The *-c Switch*

Use the –c switch to display the local NetBIOS name cache on the workstation it's running on. Figure 17.4 shows sample output of the nbtstat –c command.

FIGURE 17.4 Sample output of the nbtstat –c command



Each entry in this display shows the NetBIOS name, the hex ID for the service that was accessed, the type of NetBIOS name (unique or group), the IP address that the name resolves to, and its life. The Life value shows how many seconds each entry will live in the cache. When this time expires, the entry will be deleted.

NOTE Sometimes, displaying nbstat to display the cache will get you the response “No names in the cache”, because all entries in the cache have expired. This is what happens if you don’t regularly access machines or services with NetBIOS names.

The *-n Switch*

The –n switch will give you the local NetBIOS name table on a Windows device. Figure 17.5 shows output that’s similar to the output of the –a switch, except for one important thing: What you’re seeing is the NetBIOS name table for the machine you’re running the command on instead of that of another host. Check it out.

FIGURE 17-5 Sample output of the nbstat -n command

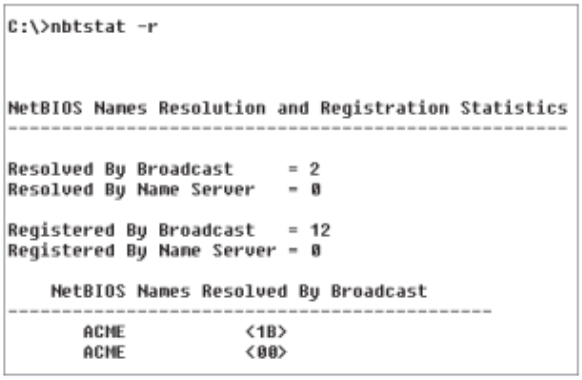
```
C:\NBTSTAT -n

Node IpAddress: [204.153.163.4] Scope Id: []

            NetBIOS Local Name Table

   Name               Type         Status
---------------------------------------------
DEFAULT        <00>  UNIQUE      Registered
WORKGROUP      <00>  GROUP       Registered
DEFAULT        <03>  UNIQUE      Registered
DEFAULT        <20>  UNIQUE      Registered
WORKGROUP      <1E>  GROUP       Registered
WORKGROUP      <1D>  UNIQUE      Registered
..__MSBROWSE__.<01>  GROUP       Registered
ADMINISTRATOR  <03>  UNIQUE      Registered
```

The *-r Switch*

This switch is probably the one you’ll use most often when you want to get hold of NetBIOS over TCP/IP (NBT) statistics, because it tells you exactly how many NetBIOS names have been resolved to TCP/IP addresses. Figure 17.6 shows sample output of the nbtstat –r command.

Figure 17.6 Sample output of the nbstat -r command

FIGURE 17.6 Sample output of the nbtstat -r command



What you can see here is that the statistics are divided into two categories. First, there are the NetBIOS Names Resolution and Registration Statistics. This is how many names have been resolved or registered either by broadcasts on the local segment or via lookup from a WINS name server. Next you have the NetBIOS unique and group names and their associated hex IDs that were resolved or registered. In Figure 17.6, you can see that there's a distinct lack of information regarding names resolved by a name server. What this means is that the output is telling you that there's no WINS server operating—instead, all NetBIOS names were resolved by broadcast only.

NOTE The –r switch comes in handy when you want to determine how a workstation is resolving NetBIOS names and whether WINS is configured correctly. If WINS isn't configured correctly or it's simply not being used, the numbers in the Resolved By Name Server and Registered By Name Server categories will always be zero.

The *-R Switch*

Unlike the –a and –A switches, -r and -R use the same letter but do not have anything in common. Here's an example. Let's say you have a bad name in the NetBIOS name cache but the right name is in the LMHOSTS file instead. (The LMHOSTS file contains NetBIOS names of stations and their associated IP addresses.) Because the cache is consulted before the LMHOSTS file is, that bad address will remain in the cache until it expires. This command is used when you want to purge the NetBIOS name table cache and reload the LMHOSTS file into memory. You do that using the nbtstat command with the –R switch, like so: nbstat -R.

You can practice this nbstat -R command on your host to purge the NBT remote cache table.

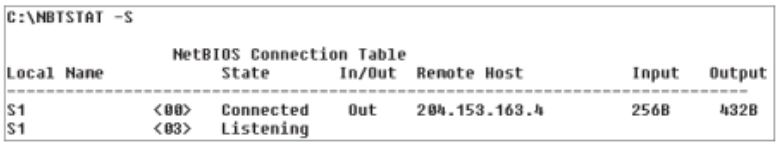
The *-S Switch*

Using the -S switch will display the NetBIOS sessions table that lists all NetBIOS sessions, incoming and outgoing, to and from the host from which you issued the command. The –S

switch displays both workstation and server sessions but lists remote addresses by IP address only. Figure 17.7 shows sample output of the nbtstat –S command.

Figure 17.7 Sample output of the nbstat -S command

FIGURE 17.7 Sample output of the nbtstat –S command



```
C:\NBTSTAT -S

                NetBIOS Connection Table
Local Name            State       In/Out  Remote Host           Input   Output
-------------------------------------------------------------------------------
S1             <00>   Connected    Out    204.153.163.4         256B     432B
S1             <03>   Listening
```

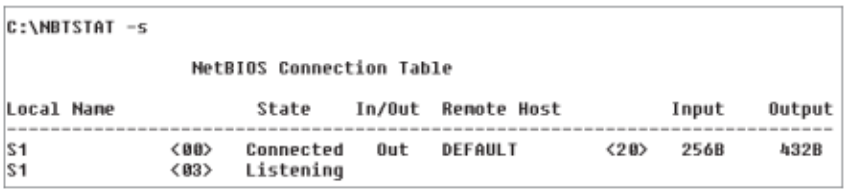
Here you can see the NetBIOS name being displayed along with its hex ID and the status of each session. An entry in the In/Out column determines whether the connection has been initiated from the computer on which you're running nbtstat (outbound) or whether another computer has initiated the connection (inbound). The numbers in the Input and Output columns indicate in bytes the amount of data transferred between the stations.

The *-s Switch*

As with the -A and the -a switches, the lowercase –s switch is similar to its uppercase sibling. The nbtstat –s command produces the same output as nbtstat –S except that it will also attempt to resolve remote-host IP addresses into host names. Figure 17.8 shows sample output from the nbtstat –s command.

Figure 17.8 Sample output of the nbstat -s command

FIGURE 17.8 Sample output of the nbtstat –s command



```
C:\NBTSTAT -s

                NetBIOS Connection Table

Local Name            State     In/Out  Remote Host          Input    Output
-------------------------------------------------------------------------------
S1             <00>   Connected   Out   DEFAULT        <20>   256B     432B
S1             <03>   Listening
```

Note the similarities between Figure 17.8 and Figure 17.7

NOTE As with any netstat command, the nbstat command can place a number for an interval at the end to direct it to deploy once every so many seconds until you press Ctrl+C.

Using the *netstat Utility*

Using netstat is a great way to check out the inbound and outbound TCP/IP connections on your machine. You can also use it to view packet statistics like how many packets have been sent and received, the number of errors, and so on.

When used without any options, netstat produces output similar to the following, which shows all the outbound TCP/IP connections. This utility is a great tool to use to determine the status of outbound web connections. Take a look:

C:/users/tlammle> netstat

Active Connections

Proto Local Address Foreign Address State TCP 10.100.10.54:49545 gnt-exchange:epmap TIME_WAIT TCP 10.100.10.54:49548 gnt-exchange:epmap TIME_WAIT TCP 10.100.10.54:49551 gnt-exchange:1151 ESTABLISHED TCP 10.100.10.54:49557 gnt-exchange:1026 ESTABLISHED TCP 10.100.10.54:49590 gnt-exchange:epmap TIME_WAIT TCP 127.0.0.1:49174 globalnet-todd:62514 ESTABLISHED TCP 127.0.0.1:62514 globalnet-todd:49174 ESTABLISHED TCP 192.168.0.6:2492 blugro2relay:2492 ESTABLISHED TCP 192.168.0.6:2492 blugro3relay:2492 ESTABLISHED TCP 192.168.0.6:49170 64.12.25.26:5190 ESTABLISHED TCP 192.168.0.6:49171 oam-d05c:5190 ESTABLISHED TCP 192.168.0.6:49473 205.128.92.124:http CLOSE_WAIT TCP 192.168.0.6:49625 64-190-251-21:ftp ESTABLISHED TCP 192.168.0.6:49628 210-11:http ESTABLISHED TCP 192.168.0.6:49629 varp1:http ESTABLISHED TCP 192.168.0.6:49630 varp1:http ESTABLISHED TCP 192.168.0.6:49631 varp1:http ESTABLISHED TCP 192.168.0.6:49632 varp1:http ESTABLISHED TCP 192.168.0.6:49635 199.93.62.125:http ESTABLISHED TCP 192.168.0.6:49636 m1:http ESTABLISHED TCP 192.168.0.6:49638 spe:http ESTABLISHED

The Proto column lists the protocol being used. You can see that I'm connected to my Exchange server and my FTP server and that I have some HTTP sessions open; by the way, all of them use TCP at the Transport layer. The Local Address column lists the source address and the source port (source socket). The Foreign Address lists the address of the destination machine (the host name if it's been resolved) plus the fact that the destination port is a TCP port. If the destination port is known, it will show up as the well-known port. In the previous output, you see http instead of port 80, and ftp instead of port 21. The State column indicates the status of each connection. This column only shows statistics for TCP connections, because User Datagram Protocol (UDP) establishes no virtual circuit to the remote device.

Usually this column indicates ESTABLISHED when a TCP connection between your computer and the destination computer has been established. All sessions eventually time out and then close, and you can see that I have all of these listed in my netstat output.

NOTE If the address of either your computer or the destination computer can be found in the HOSTS file on your computer, the destination computer's name, rather than the IP address, will show up in either the Local Address or Foreign Address column.

The output of the netstat utility depends on the switch. By using the netstat /? command we can see the options available to us.

C:\Users\tlammle> ***netstat /?***

All of the netstat switch options are listed in Table 17.7

Table 17.7 netstat Options and Descriptions

TABLE 17.7 netstat Options and Descriptions

Option	Description
-a	Displays all connections and listening ports.
-b	Displays the executable involved in creating each connection or listening port. In some cases, well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case, the executable name is in [ ] at the bottom; on top is the component it called, and so forth, until TCP/IP was reached. Note that this option can be time-consuming and will fail unless you have sufficient permissions.
-e	Displays Ethernet statistics. This may be combined with the -s option.
-f	Displays fully qualified domain names (FQDNs) for foreign addresses.
-n	Displays addresses and port numbers in numerical form.
-o	Displays the owning process ID associated with each connection.
-p *proto*	Shows connections for the protocol specified by *proto*; *proto* may be any of TCP, UDP, TCPv6, or UDPv6. If used with the -s option to display per-protocol statistics, *proto* may be any of IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
-r	Displays the routing table.

TABLE 17.7 netstat Options and Descriptions *(continued)*

Option	Description
-s	Displays per-protocol statistics. By default, statistics are shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6; the -p option may be used to specify a subset of the default.
-t	Displays the current connection offload state. Redisplays selected statistics, pausing interval seconds between each display. Press Ctrl+C to stop redisplaying statistics. If omitted, netstat will print the current configuration information once.

Simply type **netstat** followed by a space and then the particular switch you want to use. Some switches have options, but no matter what, the syntax is basically the same.

NOTE Note that with Unix-type switches, the hyphen absolutely must be included. This is common in Microsoft operating systems for TCP/IP utilities that originate from Unix systems. I'm not going to exhaustively go over each and every switch, but make sure you practice all of these on your own Windows machine.

The -a Switch

When you use the –a switch, the netstat utility displays all TCP/IP connections and all UDP connections. Figure 17.9 shows sample output produced by the netstat –a command.

Figure 17.9 Sample output of the netstat -a command

FIGURE 17.9 Sample output of the netstat –a command

```
C:\ NETSTAT -a

Active Connections

Proto   Local Address          Foreign Address        State
TCP     default:1026           204.153.163.2:80       ESTABLISHED
TCP     default:1027           204.153.163.2:80       ESTABLISHED
TCP     default:1028           204.153.163.2:80       ESTABLISHED
TCP     default:1029           204.153.163.2:80       ESTABLISHED
UDP     default:nbname         *:*
UDP     default:nbdatagram     *:*
```

The last two entries in Figure 17.9 show that the protocol is UDP and gives the source-port nicknames nbname and nbdatagram. These are the well-known port numbers of 137 and 138, respectively. These port numbers are commonly seen on networks that broadcast the NetBIOS name of a workstation on the TCP/IP network. You can tell that this is a broadcast because the destination address is listed as *:* (meaning "any address, any port").

NOTE The State column has no entry because UDP is not a connection-oriented protocol and, therefore, has no connection state.

The most common use for the –a switch is to check the status of a TCP/IP connection that appears to be hung. You can determine if the connection is simply busy or is actually hung and no longer responding.

The -e Switch

The -e switch displays a summary of all the packets that have been sent over the Network Interface Card (NIC) as of that instant. The Received and Sent columns show packets coming in as well as being sent:

```
C:\Users\tlammle>netstat -e
Interface Statistics
                           Received            Sent

Bytes                     7426841         7226953
Unicast packets             25784           35006
Non-unicast packets          1115           12548
Discards                        0               0
Errors                          0              71
Unknown protocols               0
```

You can use the -e switch to display the following categories of statistics:

Bytes The number of bytes transmitted or received because the computer was turned on. This statistic is useful for finding out if data is actually being transmitted and received, or if the network interface isn't doing anything at all.

Unicast Packets The number of packets sent from or received at this computer. To register in one of these columns, the packet must be addressed directly from one computer to another, and the computer's address must be in either the source or destination address section of the packet.

Non-Unicast Packets The number of packets that weren't directly sent from one workstation to another. For example, a broadcast packet is a non-unicast packet. The number of non-unicast packets should be smaller than the number of unicast packets. If the number of non-unicast packets is as high or higher than that of unicast packets, too many broadcast packets are being sent over your network. Definitely find the source of these packets and make any necessary adjustments to optimize performance.

Discards The number of packets that were discarded by the NIC during either transmission or reception because they weren't assembled correctly.

Errors The number of errors that occurred during transmission or reception. (These numbers may indicate problems with the network card.)

Unknown Protocols The number of received protocols that the Windows networking static couldn't interpret. This statistic only shows up in the Received column because if the computer sent them, they wouldn't be unknown, right?

Unfortunately, statistics don't mean much unless they can be colored with time information. For example, if the Errors column shows 71 errors, is that a problem? It might be if the computer has been on for only a few minutes. But 71 errors could be par for the course if the computer has been operating for several days. Unfortunately, the netstat utility doesn't have a way of indicating how much time has elapsed for these statistics.