Main content

Chapter

7

Computer Networks and Cloud Computing

Change font size Main content

Chapter Contents

Chapter Introduction

r studying this chapter, you will be able to:

Describe and compare different network technologies, including dial-up, broadband, and wireless
Explain how different kinds of networks (LAN, WLAN, WAN) are connected, and how communication works in each
Explain the importance of standards and protocols for communication among computing devices
Name the layers of the network protocol hierarchy, and describe the purpose of each layer
Name four services that computer networks provide and explain their social impact
Explain cloud computing and discuss its potential benefits

Describe the highlights of the history of the Internet and the web

Asmall font Amedium font Alarge font

Add Bookmark to this Page

help?

Main content

Chapter Contents

7.1 Introduction

Every once in a while there occurs a technological innovation of such importance that it forever changes society and the way people live, work, and communicate. The invention of the printing press by Johannes Gutenberg in the mid-15th century was one such development. The books and manuscripts it produced helped fuel the renewed interest in science, art, and literature that came to be called the Renaissance, an era that influenced Western civilization for more than 500 years. The Industrial Revolution of the 18th and early 19th centuries made consumer goods such as clothing, furniture, and cooking utensils affordable to the middle class and changed European and American societies from rural to urban and from agricultural to industrial. In the 20th century we are certainly aware of the massive social changes, both good and bad, wrought by inventions such as the telephone, automobile, airplane, television, computer, and smartphone.

We are no doubt witnessing yet another breakthrough, one with the potential to make as great a change in our lives as those just mentioned. This innovation is the *computer network*—computers connected together for the purpose of sharing personal communications, hardware and software resources, and information. During the early stages of network development, the only information exchanged was text such as email,

database records, and technical papers. However, the material sent across a network today can be virtually anything—television and radio shows, videos, music, photographs, and movies, to name just a few. If information can be represented in binary, it can be transmitted across a network.

The possibilities created by this free flow of data are enormous. Networks equalize access to information and eliminate the concept of "information haves" and "information have-nots." Students in a poorly funded rural school are no longer handicapped by an out-of-date library collection. A physician practicing in an emerging economy is able to transmit patient records, test results, and medical images to specialists anywhere in the world and have immediate access to the databases and reference works of major medical centers. Small-business owners can locate suppliers and customers on an international scale. Researchers have the same ability to communicate with experts in their discipline whether they are in New York, New Delhi, or New Guinea.

Networking can also foster the growth of democracy and global understanding by providing unrestricted access to newspapers, magazines, radio, and television, as well as supporting the unfettered exchange of diverse and competing thoughts, ideas, and opinions. However, it can also be a vehicle for spreading rumors, falsehoods, and disinformation around the world in a fraction of a second. Because we live in an increasingly information-oriented society, network technology contains the seeds of massive social and economic change. It is no surprise that during civil uprisings, political leaders who want to prevent the dissemination of opposing ideas often move quickly to restrict access to the Internet, especially social media sites.

In Chapter 6, you saw how system software can create the appearance of a user-friendly "virtual machine" on top of the raw hardware of a single computer. In today's world, computers are seldom used as isolated, standalone devices, and the modern view of a virtual machine has expanded into a worldwide collection of interconnected systems and resources. In this chapter, we take a detailed look at the underlying technology of computer networks—what they are and how they work, with a focus on the most widely used network, the Internet, and its most important application, the World Wide Web (better known now as simply "the web"). We discuss the benefits networks can bring, including one of the most advanced and powerful forms of networking: cloud computing. We conclude with a survey of how the Internet and the web came to be.

7.2 Basic Networking Concepts

A **computer network** is a set of independent computer systems interconnected by telecommunication links for the purpose of sharing information and resources. The individual computers on a network are referred to as **nodes** or *hosts*, and they can range in size from smartphones, tablets, and tiny laptops to massively parallel supercomputers. In this section, we describe some of the basic technical characteristics of a computer network.

7.2.1 Communication Links

The communication links used to build a network vary widely in physical characteristics, error rate, and transmission speed. In the approximately 50 years that networks have existed, telecommunications facilities have undergone enormous changes.

In the early days of networking, the most common way to transmit data was via **switched, dial-up telephone lines**. The term *switched, dial-up* means that when you dial a telephone number, a circuit (i.e., a path) is temporarily established between the caller and the call recipient. This circuit lasts for the duration of the call, and when you hang up it is terminated.

The voice-oriented dial-up telephone network was originally an *analog* medium. As we first explained in Chapter 4, this means that the physical quantity used to represent information, usually voltage level, is continuous and can take on any value. An example of this is shown in Figure 7.1(a). Although analog is fine for transmitting the human voice, which varies continuously in pitch and volume, a computer produces *digital* information—specifically, a sequence of 0s and 1s, as shown in Figure 7.1(b).

Figure 7.1 Two forms of information representation

For the binary signals of Figure 7.1(b) to be transmitted via a switched, dial-up telephone line, the signal must be restructured into the analog representation of Figure 7.1(a). The device that accomplishes this is a **modem**, which modulates, or alters, a standard analog signal called a *carrier wave* so that it encodes binary information. The modem modifies the physical characteristics of the carrier wave, such as amplitude or frequency, so that it is in one of two distinct states, one state representing 0 and the other state representing 1. Figure 7.2 shows how a modem can modulate the amplitude (height) of a carrier wave to encode the binary signal 1010.

Figure 7.2 Modulation of a carrier to encode binary information

At the other end of the transmission line, a modem performs the inverse operation, which is called *demodulation*. (Modem is a contraction of the two terms *mo*dulation and *dem*odulation.) It takes the received waveform, separates the carrier wave from the encoded digital signal, and passes on the digital data to the computer.

In the early days of telecommunications—the 1970s and 1980s—the **bandwidth**, or rate at which information could be sent and received, was limited to about 1,200–9,600 bits per second (bps). Advances in dial-up modem design produced devices that could transmit at 56,000 bps, or 56 Kbps, an order-of-magnitude increase. However, this is still much too slow to handle the transmission of large multimedia-based documents such as webpages, sound files, and streaming video.

Today, a technology called **broadband** has replaced modems and analog phone lines for virtually all data communications. The term *broadband* generally refers to any communication link with a transmission rate exceeding 256,000 bps. Today, most broadband links have speeds well in excess of that, often 25 million bps or more. In the case of home users, there are two widely available broadband options—digital subscriber lines (DSL) and cable modems.

A **digital subscriber line (DSL)** uses the same wires that carry regular telephone signals into your home and therefore is provided by either your local telephone company or someone certified to act as its intermediary. Although it uses the same wires, a DSL signal uses a different set of frequencies, and it transmits digital rather than analog signals. Therefore, the voice traffic generated by talking with a friend on the phone does not interfere with a webpage being simultaneously downloaded by someone else in the family. Furthermore, unlike the modem that requires that you explicitly establish a connection (dial a number) and end a connection (hang up), a DSL is a permanent "always-on" link, which eliminates the aggravating delay of dialing and waiting for the circuit to be established.

A digital subscriber line is often *asymmetric*. This means it does not have the same transmission speed in the *download* direction (from the network to your computer) as in the *upload* direction (from your computer to the network). That is because most users consume much more data than they generate. For example, to obtain a webpage, your computer sends a request message to the machine with that page. (It does this by sending the address of that page, such as www.macalester.edu.) This request message is small and contains only a few dozen characters. However, the page you receive—complete with graphics, images, and sound—could contain millions of bits. To handle this imbalance, a DSL provides greater bandwidth coming into your computer than going out. Typical DSL speeds are 5–50 million bits per second (Mbps) for downloads and 1–5 Mbps for uploads—much more than is available from a dial-up modem.

The second option for broadband communications is a **cable modem**. This technology makes use of the links that deliver cable TV signals into your home, so it is offered by cable TV providers. Some of the link capacity previously allocated for TV signals is now used for data communications. Like a DSL, a cable modem also provides an always-on

link and offers speeds generally comparable to those provided by a DSL—10 to 100 Mbps for downloads and 1–5 Mbps for uploads.

In the commercial and office environment, the most widely used broadband technology is **Ethernet**. Ethernet was developed in the mid-1970s by computer scientists at the Xerox PARC research center in Palo Alto, California. It was originally designed to operate at 10 Mbps using coaxial cable. However, 10 Mbps proved too slow for many emerging applications, so in the early 1990s researchers developed a "new and improved" version, called **Fast Ethernet**, which transmits at 100 Mbps across coaxial cable, fiber-optic cable, or regular twisted-pair copper wire.

Because even 100 Mbps may not be fast enough for multimedia applications, computer science researchers began investigating the concept of **gigabit networking**—transmission lines that support speeds of 1 billion bits per second (Gbps). In 1998 the first international **gigabit Ethernet standard** was adopted by the **IEEE (Institute of Electrical and Electronics Engineers)**, an international professional society responsible for, among other things, developing industrial standards in the area of telecommunications. The standard supports communication on an Ethernet cable at 1,000 Mbps (1 Gbps), 100 times faster than the original 10 Mbps standard. Most classrooms and office buildings today are wired to support Ethernet speeds of 1,000 Mbps—18,000 times faster than a 56K modem! In addition, virtually every desktop and laptop sold today comes with a built-in Ethernet interface, and new homes and dorm rooms are often equipped with Ethernet links.

However, not willing to rest on their laurels (and realizing that even faster networks will be needed to support future research and development), work immediately began on a new *10-gigabit Ethernet* standard, a version of Ethernet with a data rate of 10 billion bits per second. That standard was adopted by the IEEE in 2003. To get an idea of how fast that is, in a single second a 10 Gbps Ethernet network could transmit the contents of 1,700 books, each 300 pages long. In June 2010, the IEEE ratified the *100-gigabit Ethernet* standard defining a local area network that can transmit data at the almost unimaginable rate of 100 billion bits of information per second!

Do applications truly need to transmit information at billions of bits per second? To answer that question, let's determine how long it takes to transmit a high-resolution color image, such as a CAT scan, satellite image, or single movie frame, at different transmission speeds. As described in Section 4.2, a high-resolution color image contains at least 10 million picture elements (pixels), and each pixel is encoded using 8–24 bits. If we assume 16 bits per pixel, then a single uncompressed image would contain at least 160 million bits of data. If the image is compressed before it is sent, and the compression ratio is 20:1 (see Section 4.2 for a definition of compression ratio), then we must transmit a total of 8 million bits to send this single image. Figure 7.3 shows the time needed to send this amount of information at the speeds discussed in this chapter.

Figure 7.3

Transmission time of an image at different transmission speeds

Line Type	Speed	Time to Transmit 8 Million Bits (One Compressed Image)	
Dial-up phone line	56 Kbps	2.4	minutes
DSL line, cable modem	2 Mbps	4	seconds
Ethernet	10 Mbps	0.8	second
Fast Ethernet	100 Mbps	0.08	second
Gigabit Ethernet	1 Gbps	0.008	second
10-gigabit Ethernet	10 Gbps	0.0008	second
100-gigabit Ethernet	100 Gbps	0.00008	second

Figure 7.3 clearly demonstrates the need for high-speed communications to support applications such as video on demand and medical imaging. Receiving an 8 Mb image using a 56 Kbps modem takes 2.4 minutes, an agonizingly long time. That same 8 Mb image can be received in 4 seconds using a DSL or cable modem with a download speed of 2 Mbps, 0.8 second using 10 Mbps Ethernet, and a blazing 0.08 second with 100 Mbps Ethernet.

However, even 0.08 second might not be fast enough if an application requires the rapid transmission of multiple images or a huge amount of data in a short period of time. For example, to watch a real-time video image without flicker or delay, you need to stream at least 24 frames per second. Any less and the human eye notices the time delay between frames. If each frame contains 8 Mb, you need a bandwidth of . This is beyond the speed of modems, DSL, cable modems, and even 100 Mbps Ethernet, but it is achievable using gigabit networks.
An extremely important development in the field of telecommunications is the explosive growth in the use of **wireless data communication** using radio, microwave, and infrared signals. Although devices such as DSLs and cable modems provide high-speed network links, they require a user to be physically adjacent to the communication device and to have a plug and cable with the appropriate connector. This is often inconvenient or impossible.

However, in the wireless world, users' devices no longer need to be physically connected to a wired network to communicate across a network. Wireless data networks have liberated computer users just as mobile phones liberated telephone users. Using wireless, you can be sipping coffee in your favorite café, riding in a car, or working on

the factory floor and still send and receive email, access online databases, post to a social networking account, and surf the web, provided you can connect (wirelessly, of course) to a wireless network. The ability to deliver data to users regardless of their physical location is called **mobile computing**.

There are three types of wireless networks, and they are classified by the distance that the wireless signal must travel—short, medium, and long distance. In a **wireless local area network (WLAN)**, a short-distance form of wireless networking, a user transmits from his or her computer, tablet, or smartphone to a local *wireless base station*, often referred to as a *wireless router, access point*, or *hot spot*, that is no more than a few hundred feet away. This base station is then connected to a traditional wired network, such as a DSL or cable modem, to provide full Internet access. This is the type of short-distance wireless configuration typically found in a home, library, office, or coffee shop because it is cheap, simple, low powered, and easy to install. A typical local wireless configuration is shown in Figure 7.4.

Figure 7.4 Typical WLAN configuration

One of the most widely used standards for wireless local access is **Wi-Fi**, also referred to by its official name, the **IEEE 802.11 wireless network standard**. Wi-Fi is used to connect a computer to the Internet when it is within range (typically 150–300 feet or 45–90 meters) of a wireless base station, often advertised as a **Wi-Fi hot spot**. Wi-Fi systems generally use the 2.4 GHz radio band for communications and support download transmission speeds of about 10–50 Mbps. Researchers are investigating the use of higher radio frequencies to support gigabit Wi-Fi communication speeds.

Another popular short-distance wireless standard is **Bluetooth**. It is a low-power wireless standard used to communicate between devices located very close to each other, typically no more than 20–30 feet (6–10 meters) apart. Bluetooth is often used to support communication from wireless peripherals such as mice, printers, earphones, or keyboards, to a laptop or desktop system located close by. It also supports exchanges between other digital devices including mobile phones, cameras, speakers, video game consoles, and your automobile's sound system. Bluetooth is a popular technique for implementing what is termed a **personal area network (PAN)**, a collection of privately owned interconnected digital devices all located in close proximity.

A relatively new development in wireless networking is the medium-distance **metropolitan area network (MAN)**. This is a wireless network whose scope is larger than the few hundred feet of a WLAN, typically a few blocks up to an entire city. Its purpose is to provide full Internet connectivity to all computers within a neighborhood or metropolitan area. A number of cities in the United States, Europe, and Asia have installed public access Wi-Fi routers every few blocks, often on top of telephone poles or tall buildings. These routers provide convenient, low-cost wireless Internet access to all residents. The idea behind a MAN is to treat Internet services as a public utility, much like electricity, gas, and water, which is provided to individuals by a local or regional government agency.

The third wireless network category (after the short-distance LAN and the medium-distance MAN) is long-distance wireless service, called a **wireless wide area network (WWAN)**. A **wide area network** or WAN connects devices that are not in close

proximity but are across town, across the country, or across the ocean. In a WWAN, the computer (often a tablet or smartphone) transmits messages to a remote base station provided by a telecommunications company, which may be located many miles away. The base station is usually a large cellular antenna placed on top of a tower or building, providing both long-distance voice and data communication services to any system within sight of the tower. One of the most popular wide area wireless technologies is called 4G, for fourth generation technology. It offers voice services as well as data communication at rates of 50 to 500 Mbps, with peak speeds reaching 1 Gbps.

Although wireless data communication is an exciting development in computer networking, it is not without problems that must be studied and solved. For example, some forms of wireless, such as microwaves, are line of sight, traveling only in a straight line. Because of the curvature of the Earth, transmitters must be placed on top of hills or tall buildings, and they cannot be more than about 10–50 miles (15–80 kilometers) apart, depending on height. This can leave small geographical regions (often termed "dead zones") that do not have access to cellular data or voice services. Other types of wireless media suffer from environmental problems; they are strongly affected by rain and fog, cannot pass through obstacles such as buildings or large trees, and have higher error rates than wired communication. Although a few random "clicks" and "pops" do not disrupt voice communications over a mobile phone, it can be disastrous for some types of data communications. For example, if you are transmitting data at 100 million bits per second (Mbps), a breakup on the line that lasts only one one-hundredth of a second could cause the loss of one million bits of data. Although the loss of one million bits might not significantly affect your ability to enjoy an HD movie, it could have serious consequences on the transmission of financial data. Finally, there is the issue of security. Currently, it is not difficult to intercept wireless transmissions and gain unauthorized access to user messages. All of these are ongoing concerns being investigated by the computer science and telecommunications research community. However, the rapid increase in the number of mobile devices along with the ease and convenience of remote access guarantees the continuing growth and popularity of wireless data communications.

The Internet of Things

From all that we have said, you might get the idea that the Internet is solely a way for computer users, that is, people, to share computing resources or data in the form of text, images, or video. Although that was originally the case, it is no longer true. It is now possible to place uniquely identifiable network "tags" on objects as diverse as refrigerators, thermostats, automobiles, eyeglasses, TVs, pets, and machine tools. Today, any living object or mechanical entity with which we wish to communicate or to control is a candidate to be tagged.

This network tagging mechanism comes in many forms, one of the most popular being **RFID**, an acronym for a radio frequency identifier. An RFID is a communications device that has its own Internet address, the same type of network address that your laptop, tablet, or smartphone uses to communicate over the Internet. Once an RFID tag is placed on an object it can send and receive messages and is said to be *network ready*. This object can now be located and controlled by other devices on the Internet via the exchange of messages, such as the following exchange between a home thermostat and a control computer:

Thermostat → control computer: "The temperature has risen to ."
Control computer → thermostat: "Turn on air conditioning. Set to ."

The potential applications for Internet-ready devices are vast: In a business environment, inventory control becomes automatic as the company's computers can communicate directly with the stock on the shelf to determine if items need to be reordered. *Smart homes* include intelligent environmental controls that maximize comfort while minimizing cost and security systems that track what is happening inside the house to keep everyone safe—for example, turning off the oven or locking the front door if you forget. Urban planners can receive continuous streams of status information from transportation systems, utility plants, and safety offices to allow these publicly funded systems to be operated in the most cost-efficient manner.

The Internet is no longer a network of just computers and data, but also is becoming a true "Internet of Things."

Practice Problems

Show how the 4-bit digital value 0110 is converted to an analog signal by a modem that modulates the *frequency* of a carrier wave, rather than its amplitude.

Answer

The figure shows the representation of a binary signal using frequency modulation of a carrier wave.

Consider an uncompressed 1,200 × 780 image, with each pixel stored using an 8-bit grayscale representation. If we want to transmit the entire image in under 1 second, what is the minimum acceptable transmission speed?

Answer

The number of bits in the image is . To transmit this in 1 second requires a transmission speed of 7,488,000 bps, or nearly 7.5 Mbps.
How long would it take to transmit that 1,200 × 780 image from Practice Problem 2 using a 1-gigabit Ethernet network?

Answer

The number of bits in the image is .

7.2.2 Local Area Networks

A **local area network (LAN)** connects hardware devices such as computers, printers, and storage devices that are all in relatively close proximity (150–300 ft/45–90 m). (A diagram of a LAN was provided in Figure 6.20.) Examples of LANs include the interconnection of machines in one room or in the same office building. An important characteristic of a LAN is that the owner of the computers is also the owner of the means of communications. Because a LAN is located entirely on private property, the owner can

install telecommunications facilities without having to purchase services from a third-party provider such as a phone or cable company, although a third-party provider is still needed to connect the LAN to the outside world.

The previous section described how a wireless local area network can be set up using Wi-Fi and a router connected to a wired network. Here we take a look at the properties of that wired network. Wired LANs can be constructed using a number of different topologies; some of the most common are shown in Figure 7.5. In the **bus** topology, Figure 7.5(a), all nodes are connected to a single, shared communication line. If two or more nodes use the link at the same time, the messages collide and are unreadable, and, therefore, nodes must take turns using the line. The cable modem technology described in Section 7.2.1 is based on a bus topology. A number of homes are all connected to the same shared coaxial cable. If two users want to download a webpage at the exact same time, then the effective transmission rate is lower than expected because one of them must wait.

Figure 7.5 Some common LAN topologies

The **ring** topology of Figure 7.5(b) connects the network nodes in a circular fashion, with messages circulating around the ring in either a clockwise or counterclockwise direction until they reach their destination. Finally, the **star** topology, Figure 7.5(c), has a single central node that is connected to all other sites. This central node can route information directly to any other node in the LAN. Messages are first sent to the central site, which then forwards them to the correct location.

Ethernet uses the bus topology of Figure 7.5(a). To send a message, a node places the message, including the destination address, on the cable. Because the line is shared, the message is received by every other node (assuming no one else sent at the exact same time and garbled the data). Each node looks at the destination address to see if it is the intended recipient. If so, it accepts the message; if not, it discards it.

There are two ways to construct an Ethernet LAN. In the first method, called the **shared cable**, a wire (such as twisted-pair copper wire, coaxial cable, or fiber-optic cable) is literally strung around and through a building. Users tap into the cable at its nearest point using a device called a ***transceiver***, as shown in Figure 7.6(a). Because of technical constraints, an Ethernet cable has a maximum allowable length. For a large building or campus, it may be necessary to install two or more separate cables and connect them via hardware devices called repeaters or bridges.

Figure 7.6 An Ethernet LAN implemented using shared cables

A **repeater** is a device that simply amplifies and forwards a signal. In Figure 7.6(b), if the device connecting the two LANs is a repeater, then every message on LAN1 is forwarded to LAN2, and vice versa. Thus, when two Ethernet LANs are connected by a repeater, they function exactly as if they were a single network.

A **bridge** is a "smarter" device that has knowledge about the nodes located on each separate network. It examines every message to see if it should be forwarded from one

network to another. For example, if node A is sending a message to node B, both of which are on LAN1, then the bridge does nothing with the message. However, if node A on LAN1 is sending a message to node C on LAN2, then the bridge copies the message from LAN1 onto LAN2 so node C is able to see it and read it.

In the second approach to constructing an Ethernet LAN, there is no shared cable strung throughout the building. Instead, there is a box called a **switch** located in a room called a *wiring closet*. The switch contains a number of *ports*, with a wire leading from each port to an Ethernet interface in the wall of a room in the building, or to a wireless router somewhere in the building. To connect to the network, you first activate the port located in your office, typically by flipping an on/off button, and then plug your machine into the wall socket. This approach is shown in Room 101 of Figure 7.7. Alternatively, you could use Wi-Fi to transmit from your computer to a wireless router located somewhere in the building. This router would then connect to one of the Ethernet ports in the switch. This approach is shown in Room 103 of Figure 7.7. In either case, it is no longer necessary to climb into the ceiling or crawl through ductwork looking for the cable because the shared cable is located inside the switch instead of inside the building walls. That is why switches are the most widely used technique for constructing Ethernet LANs.

Figure 7.7An Ethernet LAN implemented using a switch

Practice Problems

What changes, if any, must be made to our description of the Ethernet protocol to allow a message to be sent by node A on a local area network to *every other* node on that same LAN? This operation is called **broadcasting**.

Answer

Because the message is to be broadcast, there is no need to include a destination address. Each node reads the message.

Assume you are given the following configuration of three local area networks, called LAN1, LAN2, and LAN3, connected by bridges B1 and B2.

Explain exactly how node A on LAN1 sends a message to node B on LAN3.

Answer

Node A sends the message on LAN1, where every node receives it, but only bridge B1 keeps it. All other nodes discard it because it is not addressed to them. Bridge B1 removes the message from LAN1 and rebroadcasts it on LAN2. Every node on LAN2 receives the message, but again, only bridge B2 keeps it. All others discard it. Bridge B2 knows that node B is located on LAN3, so it rebroadcasts the message on LAN3, where node B receives it, recognizes its own address, and removes the message from the network. The message has arrived at its intended destination.

What security issues could arise from the use of an Ethernet-like bus topology to implement a local area network?

Answer

The main security concern is that in an Ethernet-like bus topology every node receives a copy of every message sent. A node is supposed to look at the address field of the message to see if it is the desired recipient and, if not, discard that message. However, a malicious node could keep, read, and record every message on the network, even those that are not intended for it.

7.2.3 Wide Area Networks

As defined previously, a *wide area network* (*WAN*) connects devices that are not in close physical proximity. Because WANs cross public property, the WAN owner must purchase telecommunications services, like those described in Section 7.2.1, from an external provider. Typically, these are **dedicated point-to-point lines** or wireless links that directly connect two machines, not the shared channels found on a LAN such as Ethernet. The typical structure of a WAN is shown in Figure 7.8. This type of interconnection system in which a node is connected to other network nodes via direct links is called a **mesh network**.

Figure 7.8 Typical structure of a wide area network

Most WANs use a **store-and-forward, packet-switched** technology to deliver messages. Unlike a LAN, in which a message is broadcast on a shared channel and is received by all nodes, a WAN message must "hop" from one node to another to make its way from source to destination. The unit of transmission in a WAN is a **packet**—an information block with a fixed maximum size that is transmitted through the network as a single unit. If you send a short message, then it can usually be transmitted as a single packet. However, if you send a long message, the source node may "chop" it into *N* separate packets (such as the first 1,000 characters, the next 1,000 characters, and so on) and send each packet independently through the network. When the destination node has received all *N* packets, it reassembles them into a single message.

For example, assume the six-node WAN shown in Figure 7.9. To send a message from source node A to destination node D, the message could go from A → B → C → D. Alternately, the message may travel from A → B → F → D or A → E → F → D. The exact route is determined by the network, not the user, based on which path can deliver the message most quickly. If the message has been broken up into multiple packets, each packet might take a different route.

Figure 7.9 **Six-node WAN**

One of the nicest features of a store-and-forward network is that the failure of a single line or a single node does not necessarily bring down the entire network. For example, assume the line connecting node B to node C in Figure 7.9 crashes. Nodes B and C can still communicate via the route B → F → D → C. Similarly, if node F fails completely, nodes E and D, located on either side of F, can still exchange messages. However, instead of talking via node F, they now use the route E → A → B → C → D.

Reliability and fault tolerance were the reasons that WANs were first studied in the late 1960s and early 1970s. The U.S. military was interested in communication systems that could survive and function even if some of their components were destroyed, as might happen in a time of war or civil unrest. Their research ultimately led to the creation of the Internet. (We will have much more to say about the history of networking and the Internet later in this chapter.)

7.2.4 Overall Structure of the Internet

We have defined three classes of networks, LANs, MANs, and WANs, but all real-world networks, including the Internet, are a complex mix of all three of these network types.

For example, a company or a college would typically have one or more LANs connecting its local computers—a computer science department LAN, a humanities division LAN, an administration building LAN, and so forth. These individual LANs might then be interconnected into a private company or campus network that allows users to send email to other employees in the company and access the resources of other departments. These individual networks are interconnected via a device called a **router**. Like the bridge in Figure 7.6(b), a router transmits messages between two distinct networks. However, unlike a bridge, which connects two identical types of networks, routers can transmit information between networks that use totally different communication techniques—much as an interpreter functions between two people who speak different languages. For example, a router, not a bridge, is used to send messages from a wireless Wi-Fi network to a wired Ethernet LAN or from an Ethernet LAN to a packet-switched, store-and-forward WAN. You can see this type of interconnection structure in Figure 7.10.

Figure 7.10 **Structure of a typical company network**

The configuration in Figure 7.10 allows the employees of a company or the students of a college to communicate with each other, or to access local resources. But how do these people reach users outside their institution or access remote resources such as webpages that are not part of their own network? Furthermore, how does an individual home user who is not part of any company or college network access the larger community? The answer is that a user's individual computer or a company's private network is connected to the rest of the world through an **Internet service provider (ISP)**. An ISP is a business whose purpose is to provide access from a private network (such as a corporate or university network) to the Internet or from an individual's home computer to the Internet. This access occurs through a WAN owned by the ISP, as shown in Figure 7.11 or possibly through a MAN provided by a city or county. An ISP typically provides many ways for a user to connect to this network, from 56 Kbps modems to dedicated broadband telecommunication links with speeds in excess of billions of bits per second.

Figure 7.11 Connecting to the Internet using an ISP

The scope of networking worldwide is so vast that a single ISP cannot possibly hope to directly connect a single campus, company, or individual to every other computer in the world, just as a single airport cannot directly serve every possible destination. Therefore, ISPs (that is, ISP networks) are hierarchical, interconnecting to each other in multiple layers, or tiers, that provide ever-expanding geographic coverage. This hierarchical structure is diagrammed in Figure 7.12.

Figure 7.12 Hierarchy of Internet service providers

An individual or a company network connects to a *local ISP*, the first level in the hierarchy. This local ISP typically connects to a *regional* or *national ISP* that interconnects all local ISPs in a single geographic region or country. Finally, a regional or national ISP might connect to an *international ISP*, also called a *tier-1 network* or an *Internet backbone*, which provides global coverage. This hierarchy is similar to the standard telephone system. When you place a call to another country, your phone connects to a local switching center, which establishes a connection to a regional switching center, which establishes a connection to a national switching center. This national center has high-speed connections to similar national switching centers in other countries, which are connected, in turn, to regional and then local switches to establish a connection to the phone you are calling.

The diagram in Figure 7.12 is a pictorial representation of that enormously complex telecommunications entity we call the **Internet**. The Internet is not a single computer network; instead, it is a huge interconnected "network of networks" that includes nodes, LANs, MANs, WANs, bridges, routers, and multiple levels of ISPs.

Firewalls

In December 2013, Target, the third largest retail chain in the United States, announced that user account information, including credit and debit card data, from approximately 70 million users had been stolen by thieves who hacked into the company's network. This data breach, occurring during the height of the holiday shopping season, caused the company tens of millions of dollars in lost sales, huge amounts of negative publicity, and led to the resignation of its top corporate officers. Sadly, this type of information-based crime is becoming all too common.

For hundreds of years, the term *firewall* meant a physical barrier installed in a building or vehicle to prevent the spread of fire and smoke. That meaning has expanded in the past few years with the growth of computer networks and the increasingly important role they play in such areas as national security, health care, banking, and e-commerce. Because of networks, it is no longer possible to prevent the theft of information by stationing guards at the front door, as the perpetrators may be sitting at a keyboard thousands of miles away. This has led to a good deal of research into the concept of computer **firewalls**, hardware and software placed at the entry points to a company's private network to protect it from unauthorized access. These firewalls implement the security guidelines set up by the company, guidelines such as *address filtering*, specifying which Internet addresses to allow in and which to keep out, and *service filtering*, listing exactly which applications on your system should or should not be accessed over the network.

However, like a determined spammer who is constantly outwitting the latest and greatest spam filter, the determined network hacker is committed to learning about and breaching whatever new firewall algorithm has been developed for the marketplace. It is an ongoing game of cat and mouse and, so far, it appears that the criminal community's mouse is keeping one step ahead of the IT developers' cat.

We'll have more to say about security issues in Chapter 8.

> As of mid-2017, the Internet contained more than 1.06 billion nodes (hosts) and hundreds of thousands of networks located in more than 230 countries (data from http://ftp.isc.org/www/survey/reports/current/). A graph of the number of host computers on the Internet from 1994 is shown in Figure 7.13. (This figure is really an undercount because there are numerous computers located behind protective firewalls that will not respond to any external attempts to be counted.)

Figure 7.13 Internet domain survey host count graph

Source: Internet Systems Consortium (www.isc.org/services/survey)

> How does something as massive as the Internet actually work? How is it possible to get 1 billion machines around the world to function efficiently as a single system? We answer that important question in the next section.

7.3 Communication Protocols

When you talk on the telephone, there is an accepted set of procedures that you follow. For example, when you answer the phone, you say "Hello," and then wait for the individual on the other end to respond. The conversation continues until someone says "Goodbye," at which time both parties hang up. You might call this "telephone etiquette"—the conventions that allow orderly exchanges to take place. Imagine what would happen if someone were unaware of them. Such a person might pick up the phone but not say anything. Hearing silence, the caller would be totally confused, think the call did not get through, and hang up.

Similar etiquette applies to computer networks. To have meaningful communications, we need a set of procedures that specifies how the exchanges will take place. This "network etiquette" is achieved by means of network protocols.

In networking, a **protocol** is a mutually agreed-upon set of rules, conventions, and agreements for the efficient and orderly exchange of information. Even though the Internet has more than 1 billion host machines made by dozens of different manufacturers and located in hundreds of countries, they can all exchange messages correctly and efficiently for one simple reason: They have all agreed to use the same protocols to govern that exchange.

You might think that something as massive and global as the Internet would be managed by either the governments of the major industrialized nations or an international agency like the United Nations. In fact, the Internet is operated by the **Internet Society**, a nonprofit, nongovernmental, professional society composed of 145 worldwide organizations (foundations, government agencies, educational institutions, companies) along with 80,000 individual members in 100 countries united by the common goal of maintaining the viability and health of the Internet. This group, along with its subcommittees, the Internet Architecture Board (IAB) and the Internet Engineering Task Force (IETF), establishes and enforces network protocol standards. (Perhaps the fact that the Internet developed outside the scope of government bureaucracies and their "red tape" is exactly what has allowed it to become so enormously successful!) To learn more about the Internet Society and its many activities, check out its home page at www.internetsociety.org/.

The protocols that govern the operation of the Internet are set up as a multilayered hierarchy, with each layer addressing one aspect of the overall communications task. They are structured in this way because of the volatility of telecommunications and networking. By dividing the protocols into separate, independent layers, a change to the operation of any one layer will not necessarily cause a change to other layers, making maintenance of the Internet much easier.

The Internet **protocol hierarchy**, also called a protocol stack, has five layers, and their names and some examples are listed in Figure 7.14. This hierarchy is also referred to as **TCP/IP**, after the names of two of its most important protocols.

Figure 7.14 The five-layer TCP/IP protocol hierarchy

In the following sections, we briefly describe the responsibilities of each of the five layers in the hierarchy shown in Figure 7.14. Main content

Chapter Contents

7.3.1 Physical Layer

The **Physical layer protocols** govern the exchange of binary digits (bits) across a physical communication channel, such as a fiber-optic cable, copper wire, or wireless radio channel. These protocols specify such things as:

- How we know when a bit is present on the line
- How much time the bit will remain on the line
- Whether the bit is in the form of a digital or an analog signal
- The physical quantities used to represent a binary 0 and a binary 1
- The shape of the physical connector between the computer and transmission line

The goal of the Physical layer is to create a "bit pipe" between two computers, such that bits put into the pipe at one end can be read and understood by the computer located at the other end, as shown in Figure 7.15.

Figure 7.15 The concept of a bit pipe

Once you select a standardized Physical layer protocol by purchasing a cable modem, getting a DSL, or using a mobile phone with Wi-Fi capabilities, you can transmit binary signals across a physical channel. From this point on in the protocol stack, you no longer need to be concerned about such engineering issues as voltage levels, wavelengths, or radio frequencies. These details are hidden inside the Physical layer, which provides all of the necessary bit transmission services. From now on, all you need to know about the communication channel is that when you ask the Physical layer to send a bit, it does so, and when you ask the Physical layer to get a bit, it retrieves a 0 or a 1.

Main content

Chapter Contents

7.3.2 Data Link Layer

The Physical layer protocols create a bit pipe between two machines connected by a communication link. However, this link is not an error-free channel, and due to interference or weather or any number of other factors, errors can be introduced into the transmitted bit stream. The bits that come out might not be an exact copy of the bits that went in. This creates what is called the **error detection and correction** problem—how do we detect when errors occur, and how do we correct them?

Also, because we want to receive complete messages, not just raw streams of bits, we need to know which bits in the incoming stream belong together; that is, we need to identify the start and the end of a message. This is called **framing**. It is the job of the

Data Link protocols to address and solve these two issues—error handling and framing. This process is done in two stages called Layer 2a, *Medium Access Control*, and Layer 2b, *Logical Link Control*. Together these two services form the Layer 2 protocol called the Data Link layer.

In Section 7.2.1, we described how local area networks like Ethernet networks communicate by having multiple machines connected to a single, shared communication line (Figures 7.6 and 7.7). However, although shared by many machines, at any single point in time, this line is capable of sending and receiving only a single message. Attempting to send two or more messages at the same time results in all messages being garbled and none getting through. In this environment, a necessary first step in transmitting a message is determining how to allocate this shared line among the competing machines. The **Medium Access Control protocols** determine how to arbitrate ownership of a shared communication line when multiple nodes want to send messages at the same time.

This could be done in a *centralized* manner by creating a single master control node responsible for determining who gets ownership of the line at any instant in time. Although easy to do, centralized control is rarely used. One reason is that it can be slow. Each node sends its request to the master, who must decide which node gets the line, and then inform every other node of its decision. This arbitration process takes a good deal of time, making the network highly inefficient. Another problem is that centralized control is not fault tolerant. If the master node fails, the entire network is inoperable.

Most Medium Access Control protocols, including Ethernet, use a *contention-based* approach in which there is no central authority and all nodes compete equally for ownership of the line. When a node wants to send a message, it first listens to the line to see whether or not it is currently in use. If the line is idle, then the node transmits immediately. If the line is busy, the node wanting to send monitors the status of the line and, as soon as it becomes idle, it transmits. This situation is diagrammed in Figure 7.16(a), in which node B wants to send but notices that A is using the line. B listens and waits until A is finished, and as soon as that occurs, B is free to send.

Figure 7.16 The Medium Access Control protocols in Ethernet

However, there is still a problem. If two or more users want to send a message while the line is in use, then both are monitoring its status. As soon as the line is idle, both transmit at exactly the same time. This is called a **collision**, and it is a common occurrence in contention-based networks like Ethernet. When a collision occurs, all information is lost. This scenario is shown in Figure 7.16(b). According to the Ethernet protocols, when a collision occurs, the colliding nodes immediately stop sending, wait a random amount of time, and then attempt to resend. Because it is unlikely that both nodes will select the exact same random waiting period, one of them should be able to acquire the line and transmit while the other node waits a little longer. (If they do choose exactly the same random waiting time then there will be a second collision, and the two nodes will have to go through the random wait and retransmit process again.) This situation is diagrammed in Figure 7.16(c).

One reason the Ethernet protocol is so popular is that control is *distributed*. Responsibility for network operation is shared by all nodes in the network rather than centralized in a single master controller. Each node makes its own decisions about when to listen, when to send, and when to wait. That means that the failure of one node does not affect the operation of any other node in the network.

If your network uses point-to-point links like those in Figure 7.8, rather than shared lines, you do not need the Medium Access Control protocols just described because any two machines along the path are connected by a dedicated line. Therefore, regardless of whether you are using a shared channel or a point-to-point link, you now have a sender and a receiver, who want to exchange a single message, and these two nodes are directly connected by a channel. It is the job of the Layer 2b **Logical Link Control protocols** to solve the error detection and correction problem and ensure that the message traveling across this channel from source to destination arrives correctly.

How is it possible to turn an inherently error-prone bit pipe like the one in Figure 7.15 into an error-free channel? In fact, we cannot eliminate errors, but we can detect that an error has occurred and retransmit a new and unblemished copy of the original message. The **ARQ algorithm**, for *a*utomatic *r*epeat re*q*uest, is the basis for all Data Link Control protocols in current use.

Remember that at this point, nodes A and B are directly connected by a physical link. When A wants to send a message to B, it first adds some additional information to form a packet. It inserts a sequence number (1, 2, 3, ...) uniquely identifying this packet, and it adds some error-checking bits that allow B to determine if the packet was corrupted during transmission. Finally, it adds a start of packet (SOP) and end of packet (EOP) delimiter to allow node B to determine exactly where the packet begins and ends.

Thus, the packet M sent from A to B looks something like Figure 7.17. This packet is sent across the communication channel, bit by bit, using the services of the Physical layer protocols described in the previous section. When B receives the packet, it examines the error-check field to determine if the packet was transmitted correctly.

Figure 7.17A message packet sent by the Data Link protocols

What makes the ARQ algorithm work is that the sender, node A, maintains a *copy* of the packet after it has been sent. If B correctly receives the packet, it returns to A a special **acknowledgment message**, abbreviated ACK, containing the sequence number of the correctly received packet. Node A now knows that this packet was correctly received and can discard its local copy. It is free to send the next message:

A	*B*	
M (1) →		Send the first packet from A to B
	← ACK (1)	B says to A, "I got it," A can discard it
M (2) →		Send the second packet from A to B
	← ACK (2)	B says to A, "I got it," A can discard it

⋮

If B does not correctly receive the packet (or the packet is lost entirely), then A will not receive the ACK message from B. After waiting a reasonable amount of time (typically two or three times the average round trip delivery time), A resends the message to B using the copy stored in its memory:

A	*B*	
M (1) →		Send the first packet from A to B
	← ACK (1)	B says to A, "I got it," A can discard it
M (2) →		Send the second packet from A to B
M (2) →		No response from B; wait for a while and resend the second packet from A to B
	← ACK (2)	B says to A, "I got it," A can discard it

⋮

The ACK for a correctly received packet is itself a message and can be lost or damaged during transmission. If an ACK is lost, then A incorrectly assumes that the original packet was lost and retransmits it. However, B knows this is a duplicate because it has the same sequence number as the packet received earlier. It simply acknowledges the duplicate and discards it. This ARQ algorithm guarantees that every message sent (eventually) arrives at the destination.

Thus, we can think of the Data Link layer protocols as creating an error-free "message pipe," in which messages go in one end and always come out the other end correctly and in the proper sequence, as shown in Figure 7.18.

Figure 7.18 Data Link protocols create an error-free "message pipe"

Practice Problems

Node A and node B are exchanging messages using the ARQ algorithm described in this section. State what action node B should take in each of the following situations:

Node A		*Node B*
M (3)	→	
	←	ACK (3)

M (3) →

?

2. Answer

3. A has resent M(3), presumably because B's ACK(3) message never reached A. When B receives the second copy of M(3), it should again send the ACK(3) to A but discard the message because it is a duplicate.

M (3) →

← ACK (3)

M (4) →

?

5. Answer

6. B should disassemble M(4) and check that it was correctly transmitted. If so, B should send an ACK(4) message to A; if not, B should discard the faulty message and wait for A to resend it.

M (3) →

← ACK (3)

M (5) →

?

8. Answer

9. B should disassemble M(5) and check that it was correctly transmitted. If so, B should send an ACK(5) message to A; if not, B should discard the faulty message and wait for A to resend it. Although this message has been received out of sequence (presumably because M(4) was lost), no special action is required at this time; B did not send an ACK(4) so A should in time resend M(4).

9. State why the following statement is true or false: Using a contention protocol like the Ethernet you can be guaranteed that a message will eventually arrive correctly at its intended destination.

Answer

This statement is false. Two nodes can have a collision and then pick the same random waiting time. There will then be a second collision, and it is possible that they will again pick the same waiting time, leading to a third collision. Theoretically, it is possible to have an infinite number of collisions, although the probability of that happening is vanishingly small.

11. Explain why message collisions would or would not occur on local area networks that used the ring topology of Figure 7.5(b) or the star topology of Figure 7.5(c).

Answer

Collisions can occur in a ring topology because all nodes share the ring and a node may want to send a message at the same time that a message from another node is passing by. In a star topology, collisions are avoided because each node has a direct line (via the hub node) to any other node.

2. Given a 100 Mbps Ethernet network supporting approximately 20 users sending messages that average 100 characters in length, do you think that collisions would or would not be a significant problem? Explain your reasoning.

Answer

Sending a 100-character message, 800 bits, across a 100 million bit per second (100 Mpbs) Ethernet takes about seconds, or 8 μsec. The likelihood that one of the other 19 users on the system would transmit during that tiny window of time is quite small, so the probability of a collision would be small.

Main content

Chapter Contents

7.3.3 Network Layer

The first two layers of the protocol stack enable us to transmit messages from node A to node B, but only if these two nodes are directly connected by a physical link. If we look back at the model of a wide area network shown in Figure 7.8, we see that the great majority of nodes are *not* directly connected. It is the job of the end-to-end **Network layer protocols** to deliver a message from the site where it was created to its ultimate destination. To accomplish this delivery task, every node must agree to use the same addressing scheme so that everyone is able to identify that ultimate destination. Thus, the two critical responsibilities of the Network layer are as follows:

- Creating a universal addressing scheme for all network nodes; and
- Delivering messages between any two nodes in the network.

Every node in the network must run the identical Network layer protocol, and it is one of the most important parts of the protocol stack. It is often said that the Network layer is the "glue" that holds the entire network together. The Network layer in the Internet is called **IP**, for Internet Protocol.

You have almost certainly been exposed to the host naming scheme used by the Internet, as you use it in all your email and web applications. For example, the machines of the two authors of this book have the following names:

macalester.edu

iupui.edu

However, these symbolic **host names** are not the actual names that nodes use to identify each other in IP. Instead, nodes identify each other using a 32-bit **IP address**, often written as four 8-bit numeric quantities in the range 0–255, each grouping separated by a dot. For example, the machine referred to as macalester.edu has the 32-bit IP address 141.140.1.5. In binary, it appears as follows:

10001101	10001100	00000001	00000101
141	140	1	5

and this is the actual destination address placed inside a message as it makes its way through the Internet. Looking at the numeric address shown above, it is easy to understand why people prefer symbolic names. Whereas it is easy for humans to remember mnemonic character strings, imagine having to remember a sequence of 32 binary digits. (This is reminiscent of the benefits of assembly language over machine language.)

It is the task of a special Internet application called the **Domain Name System (DNS)** to convert from a symbolic host name such as macalester.edu to its equivalent 32-bit IP address 141.140.1.5. The DNS is a massive database, distributed over literally thousands of machines that, in total, contain the host name-to-IP address mappings for the more than 1 billion host computers on the Internet. When you use a symbolic host name, such as mySchool.edu or myCompany.com, this character string is forwarded to a computer called a local name server that checks to see if it has a data record containing the numeric IP address for this symbolic name. If so, it returns the corresponding 32-bit binary value. If not, the local name server forwards it to a remote name server (and possibly another, and another, …) until it locates the name server that knows the correct IP address.

I Can't Believe We've Run Out

When the Internet was started in the early 1970s, its designers selected a 32-bit addressing scheme believing, quite reasonably, that the available addresses would last for a long, long time. Well, that long, long time unexpectedly arrived on February 4, 2011 when the Internet Corporation for Assigned Names and Numbers (ICANN) announced that it had just handed out the very last unused block of network addresses. It had run out!
Fortunately, this is not yet a catastrophic situation because the careful recovery and reallocation of addresses previously handed out but never actually assigned will allow the current addressing scheme to continue functioning for quite a few more years. (As of mid-2016 there were still about 8–10 million network addresses available for assignment.) However, as ICANN likes to say, "four billion addresses is simply not enough for seven billion people." In addition, there are hundreds of millions of devices that have been assigned a network IP address as discussed in the Special Interest Box "The Internet of Things" earlier in this chapter. For that reason, Internet engineers have created a "next generation" set of IP protocols, called IPv6.

The transition from the current IPv4, with its 32-bit address, to IPv6, with a 128-bit address field, will certainly solve the "lack of addresses" problem. The new Internet protocol provides for unique addresses—that's a 3 followed by 36 zeros. The designers were determined never to run out of addresses again! However, because of the time and expense involved in converting to the new 128-bit IPv6 addresses, as of 2017, a significant number of communication companies and ISPs had not switched to this new addressing scheme, and it will likely still be a few years before the final conversion to IPv6 becomes a reality.

Let's use the diagram shown earlier to see how the Network layer operates:

Assume A wants to send a message to D. First, node A uses the DNS to obtain the 32-bit IP address of node D, which it inserts into its message. Because there is no direct path from A to D, the message is sent along a multi-hop path reaching from A to D. (Each of these direct machine-to-machine hops uses the Data Link layer protocols described in

the previous section.) In this example, there are four possibilities—ABCD, AEFD, ABFD, and AEFBCD—and the process of selecting one specific path is called **routing**.

Routing algorithms are highly complex because of the massive volume of data that must be maintained and the enormous amount of processing required to determine the optimal route, called the *shortest path*. The shortest path between two nodes is not necessarily the shortest path in length, but the path via which the message can travel the fastest (maybe it should be called the *quickest path*). To determine the shortest path between every pair of nodes, we need to know the time delay between every connected pair of nodes in the network. In the previous example, this is the time to get from A to B, from B to C, from A to E, and so on. For small networks, it is feasible to have all this data available at any one time, but for networks like the Internet, with its billions of nodes and links, this is an unimaginably huge amount of data to obtain and keep current.

Even if we were somehow able to collect all this data, we are still not finished. Now we must determine exactly which path to select. One possible algorithm is to determine the time required to send a message along *every* path from a source to a destination and then pick the one with the smallest delay. For example, to determine the optimal path from A to D, we could start out by summing the individual delays from A to B, B to C, and C to D, which would give us the time to get from A to D using the route A → B → C → D. We now repeat this process for every other path from A to D and pick the smallest.

However, in Section 3.5, we showed that, as the number of network nodes increases, the solution time for these "brute force" algorithms grows exponentially. Therefore, this method is infeasible for any but the tiniest networks. Fortunately, there are much better algorithms that can solve this problem in time, where *N* is the number of nodes in the network. (The Internet uses a method called *Dijkstra's shortest path algorithm*.) For large networks, where or , an algorithm might require on the order of or calculations to determine the best route from any node to another—still an enormous amount of work.
There are additional problems that make network routing so difficult. One complication is *topological change*. The Internet is highly dynamic, with new links and new nodes added on a daily basis. Therefore, a route that is optimal now may not be optimal in a couple of days or even a couple of hours. For example, the optimal route from A to D in our diagram may currently be A → B → C → D. However, if a new high-speed line is added connecting nodes E and D, this might change the shortest path to A → E → D. Because of frequent changes, routing tables must be recomputed often.

There is also the question of *network failures*. It may be that when everything is working properly, the optimal route from A to D is A → B → C → D. But what if node B fails? Rather than have all communications between A and D suspended, it would be preferable for the network to switch to an alternative route that does not pass through node B, such as A → E → F → D. This ability to dynamically reroute messages allows a WAN to continue operating even in the presence of node and link failures.

The Network layer has many other responsibilities not mentioned here, including network management, broadcasting, and locating mobile nodes that move around the network. The Network layer is truly a complex piece of software.

With the addition of the Network layer to our protocol stack, we no longer have just a bit pipe or a message pipe, but a true end-to-end "network delivery service" in which

messages are delivered between any two nodes in the network, regardless of their location, as illustrated in Figure 7.19.

Figure 7.19 Network layer "delivery service"

Main content

Chapter Contents

7.3.4 Transport Layer

Imagine that 123 Main St. is a large, multistory office building with thousands of tenants. When you address a letter to an employee who works in this building, it is not enough to write:

Joe Smith

123 Main St.

My Town, Minnesota

Practice Problems

Given the following six-node wide area network for which the numbers attached to the links are a measure of the "delay" in using that link (e.g., some lines could be more heavily used than others and therefore have a longer wait time), answer the following questions:

What is the shortest path from node A to node D, where *shortest path* is defined as the path with the smallest sum of the delays on each individual link? Explain exactly how you went about finding that path.

Answer

There are four distinct paths from node A to node D, and their total weights are

So the shortest path is ABFD, found by computing the weight of every possible path and then picking the smallest. This is essentially a "brute force" approach to the problem.

Do you think the algorithm you used in Practice Problem 1 works if we redraw the graph so it has 26 nodes rather than 6 and about 50 links rather than 10? Why or why not?

Answer

This approach would not work with larger graphs such as one with 26 nodes and 50 links. The number of possible paths would grow much too large for us to enumerate and evaluate them all in a reasonable amount of time. We must use a more clever algorithm.

What if the link connecting node F to node D fails? What is now the shortest path from node A to node D? Could the failure of any single link in this network prevent nodes A and D from communicating?

Answer

If the link connecting node F to node D fails, then the paths ABFD and AEFD will not work and their weights become “infinite.” Of the two paths remaining, path ABCD with weight 16 now becomes the shortest path. No one link in the network will disconnect nodes A and D. We can see that clearly by noting that the two paths ABCD and AEFD do not share any links in common. Therefore, if a link along one of these paths fails, we can use the other path.

What if the delay from node A to node E were reduced from six units to four? Would that change the shortest path from A to D?

Answer

Yes, the shortest path would change. The shortest path from A to E would now be the route AEFD with a cost of units. It had been ABFD with a cost of units.
If node B crashed and was unable to send or receive messages, would there be any other nodes in the network that would be unable to communicate?

Answer

If node B failed, no other nodes in the network would be brought down, although their communication delays might increase. Nodes A, C, D, E, F could all send and receive messages from the other active nodes.

> This identifies the correct building, but how do the people in the central mailroom locate “Joe Smith” from among the thousands of people who work there? We need to provide a more descriptive address, one that not only identifies the correct building but also exactly where inside this building Mr. Smith works:

Joe Smith

Acme Services Inc., Suite 2701

123 Main St.

My Town, Minnesota

> The same situation exists on the Internet. Every host computer has an IP address that uniquely identifies it. However, there may be multiple application programs running on that one machine, each one “doing its own thing.” When a message comes in, how do we know which application program it is for and where to deliver it?
>
> We need a second level of address that identifies not only a specific machine but also a specific program running on that machine. This “ program identifier,” usually just a small integer value, is called a **port number**, and it serves the same role as the address line “Acme Services Inc., Suite 2701.” Assigning port numbers to programs and remembering which program goes with which port is a part of the **Transport layer protocols**. Although each host computer has one IP address, it may at any instant in time have many active ports.
>
> The relationship between these two address types is shown in Figure 7.20. This diagram shows two hosts: Host A, whose IP address is 101.102.103.104, and Host B, with IP address 105.106.107.108. Host A is currently running two programs, called W and X (perhaps a web browser and an email client), with port numbers 12 and 567, respectively, while Host B is executing two programs, named Y and Z, with port numbers 44 and 709, respectively.

Figure 7.20 Relationship between IP addresses and port numbers

The Transport layer protocols create a "program-to-program" delivery service, in which we don't simply move messages from one host to another, but from a specific program at the source to a specific program at the destination.

In the example in Figure 7.20, it is the job of the Network layer protocol to deliver the message from the host with IP address 101.102.103.104 to the host with IP address 105.106.107.108, at which point its responsibilities are over. The Transport protocol at the destination node examines the newly arrived message to determine which program should get it, based on the port number field inside the message. For example, if the port number field is 709, then the information in the message is forwarded to application program Z. (What program Z does with this information and exactly what that message means are not part of the Transport protocols but rather the Application protocols discussed in the following section.)

How does a program (such as W or X) learn the port number of another program (such as Y or Z) running on a remote machine somewhere out in the network? The answer is that all important applications on the Internet use *well-known port numbers*. Just as it is widely known in the United States that directory assistance is found at 555-1212 and police and fire emergencies are reported to 911, fixed integer values are assigned to certain applications, and those values are made known to every machine on the Internet. For example, the HTTP protocol, which allows us to access remote webpages (and which we discuss in the following section), always uses port 80 for its communications. If you want to get a webpage from another machine, you simply talk to the program on that remote machine that is listening for messages on port 80.

Figure 7.22 later in the chapter lists the port numbers of some common Internet applications. A database of all well-known port assignments is maintained online by IANA, the Internet Assigned Numbers Authority.* The only time you need to get a new port number is when you are developing a new application program.

Figure 7.22

Some popular application protocols on the Internet

Acronym	Name	Application	Well-Known Port
HTTP	Hypertext Transfer Protocol	Accessing webpages	80
SMTP	Simple Mail Transfer Protocol	Sending email	25
POP3	Post Office Protocol	Receiving email	110
IMAP	Internet Message Access Protocol	Receiving email	143

Acronym	Name	Application	Well-Known Port
FTP	File Transfer Protocol	Accessing remote files	21
TELNET	Terminal Emulation Protocol	Accessing remote terminals	23
DNS	Domain Name System	Translating symbolic host names to IP addresses	42

The other primary responsibility of the Transport layer has to do with errors and reliability. When we introduced the Data Link layer in Section 7.3.2, we said that one of its tasks is to take the inherently unreliable physical channel underneath it and turn it into an efficient and error-free channel. That same type of relationship exists between the Transport layer and the layer underneath it, namely, the Network layer.

The Network layer of the Internet, IP, is an inherently unreliable communication channel. IP uses what is called a *good faith* transmission model. That means that it tries very hard to deliver a message from source to destination, but it does not guarantee either correct or ordered delivery. In this sense, IP is like the post office. The post office does a very good job of delivering mail, and the overwhelming majority of letters do get through. However, the post office does not guarantee that every letter you send will arrive, and it does not guarantee that letters will arrive either within a specific time period or in exactly the same order that they were originally posted. If you need these features, you have to use some type of "special handling" service such as registered mail or express mail.

In a sense, the Transport layer represents just this type of "special handling" service. Its job is to create a high-quality, error-free, order-preserving, end-to-end delivery service on top of the unreliable delivery services provided by IP. On the Internet, the primary transport protocol is **TCP (Transport Control Protocol)**. (There is another transport protocol called *UDP* for *User Datagram Protocol* that does not provide such a high level of service. We will not be discussing it here.)

TCP requires that the two programs at the source and destination node initially establish a connection. That is, they must first inform each other of the impending message exchange, and they must describe the "quality of service" they want to receive. This connection does not exist in a hardware sense—there is no "wire" stretched between the two nodes. Instead, it is a logical connection that exists only as entries in tables. However, TCP can make this logical connection behave exactly as if there were a real connection between these two programs. This logical view of a TCP connection is shown in Figure 7.21.

Figure 7.21 Logical view of a TCP connection

Once this connection has been established, messages can be transmitted from the source program to the destination program. Programs P1 and P2 appear to have a direct, error-free link between them. In reality, however, their communications travel via the Network layer protocol from P1 to A, B, C, D, and finally to P2, using the services of the Data Link protocol for each link along the way.

TCP uses the same ARQ algorithm described in our discussion of the Data Link level. The receiving program must acknowledge every message correctly received. If a message is lost in the network and does not arrive, the sending program does not receive an acknowledgment and eventually resends it. Every message is ultimately delivered to the application program waiting for it and, therefore, this TCP connection does function like an error-free channel.

Every message sent on this TCP connection contains a sequence number—1, 2, 3, ... If messages are received out of order (say message 3 comes in before message 2 because of errors along the route), then TCP simply holds the later message (message 3) until the earlier message (message 2) correctly arrives. At that time, it can deliver both messages to the application program in the proper order. From the destination's point of view, this TCP connection always delivers messages in the proper order.

With these four protocol layers in place, we now have a complete end-to-end delivery service. The network can transmit a message from a program anywhere in the network to another program anywhere in the network—and can do it correctly, efficiently, and in the proper order. The only thing left to specify is the *content* of those messages; that is, what does a program want to say to another program? Essentially we are asking the question: "What types of applications do we want to give to our network users and exactly how do we implement them?" We answer that fundamental question as we look at the very top layer of our protocol stack—the Application layer.

Main content

Chapter Contents

7.3.5 Application Layer

The **Application layer protocols** are the rules for implementing the end-user services provided by a network, and they are built on top of the four protocol layers described in previous sections. These services are the reason that networks exist in the first place, and the appearance of globally popular applications (often called *killer apps*) has fueled the rapid growth of networking and the Internet—email in the 1970s, chat rooms in the 1980s, the web and ecommerce in the 1990s, and social networking in the 21st century. Figure 7.22 lists a few of the important application protocols on the Internet upon which these killer apps are built.

It is not possible in this one section to discuss all the protocols listed in Figure 7.22. Instead, we will use the HTTP protocol, which is used by the web to access and deliver webpages, to serve as a general model for how Application layer services are typically built on top of the TCP/IP protocol stack.

A single webpage is identified by a symbolic string called a **Uniform Resource Locator**, abbreviated URL. URLs have three parts, and they look like this:

protocol:// host address/page

The first part, *protocol*, indicates the type of information contained in this webpage. The most common format is *hypertext*, and we access it using the **Hypertext Transfer Protocol (HTTP)**. The second part of the URL is the *host address* of the machine where the page is stored. This is the symbolic host name first discussed in Section 7.3.3. The third and last part of the URL is the *page* identification, which is usually a file stored on the specified machine. Thus, a typical URL might look like the following:

http://www.macalester.edu/about

This identifies a hypertext ("http") document stored in a file called *about* located on a host computer whose symbolic name is www.macalester.edu. (*Note*: "http" is the default protocol. Thus, the previous URL can also be written as www.macalester.edu/about.)

Before you can transfer a webpage, you must first establish a connection between the client program (the web browser run by the user) and port 80, the port number of the web server located at the node where the webpage resides, in this example www.macalester.edu. The network uses the TCP protocol described in Section 7.3.4 to establish this connection. Thus, you can clearly see how the HTTP application protocol is built on top of the TCP/IP protocol stack just described.

Once you have established the TCP connection, an HTTP *request message* is sent from the client to the server, specifying the name of the webpage that you wish to access. A second HTTP message type, called a *response message*, is returned from the server to the client along the same TCP connection. The response contains a status code specifying whether or not the request was successful and, if it was, it includes a copy of the requested page.*

Let's illustrate how these pieces work together using a simple example. Imagine you are using a web browser such as Chrome, Safari, or Firefox, and have just clicked on, touched, or entered the following URL:

http://www.macalester.edu/about

The following sequence of events now takes place:

1. Your browser scans the URL and extracts the host name of the machine to which it must connect—www.macalester.edu. (Let's disregard the issue of how this symbolic name is converted to its corresponding 32-bit or 128-bit binary IP address.)
2. Your browser asks TCP to establish a connection between itself and port 80 (the web server) of the machine called www.macalester.edu.
3. When the TCP connection between your browser and the web server is established, your browser scans the URL to identify the page you want to access. In this case, it is */about*. Your browser constructs an HTTP 'GET' message, which requests the contents of that webpage. This GET message looks something like the following:

   GET /about HTTP/1.2

   Host: www.macalester.edu

   This message says that we want a copy of the webpage */about* located at www.macalester.edu, and it should be accessed using the http protocol, version 1.2.

(An actual GET message is quite a bit more complex and includes a number of additional fields not shown here.)

4. The GET message in Step 3 is transmitted across the Internet from the client's web browser port at the source node to the web server port at the destination node using the services of TCP/IP as well as the Data Link and Physical layer protocols.
5. When the GET message arrives at the destination, it is delivered to the web server (which is listening on port 80). The web server locates the file named in the GET message (*/about*) and creates a response message containing a copy of the contents of that file. This response message looks something like the following:

   HTTP/1.2 200

   Connection: close

   Date: Monday, 26 Mar 2018

   Content Length: 53908

   Content Type: text/html

   … (the actual contents of the webpage go here) …

   This response message says that the server successfully found the file (code 200), and it contains 53,908 bytes of text. It also says that after the webpage has been sent, the TCP connection between the browser and the server will be closed. Finally, there is a copy of the entire webpage. (Again, some fields in the response message have been omitted for clarity.)

6. The HTTP response message in Step 5 is transmitted across the Internet from the web server back to the port of the client's web browser using the services of TCP/IP as well as the Data Link and Physical layer protocols.
7. The message is delivered to your web browser, and the page is displayed on the screen. The TCP connection between the two programs is terminated.
   Something similar to this occurs every time you click on a new URL. This sequence of events is diagrammed in Figure 7.23.

Figure 7.23 Behavior of the HTTP Application layer protocol

Main content

Chapter Contents

7.4 Network Services and Benefits

At the beginning of this chapter, we said that networks have the potential to create enormous social change. Now that we have looked at how they are designed and built, let's briefly examine the services they can offer and their impact on society.

Main content

Chapter Contents

7.4.1 Interpersonal Communications

Email (electronic mail) has been the single most popular application of networks for the last 35 years. It is estimated that about 400 billion email messages are transmitted across the Internet *every day*! When the Internet was first developed, its designers thought it would be an ideal way for scientists and engineers to access important software packages and data files stored on remote computers. However, the first Internet "killer app" was rather unexpected and something quite different—email.

Email is *convenient*. You can send a message whenever you want, and it waits for the recipient to log on and read at his or her convenience. Email is *fast*. A message from the United States typically arrives anywhere in the world in a few seconds, even though it may have to pass through 15 or 20 nodes along the way (using the packet-switched protocols described in the previous section). Email supports *multimedia*. The contents of your electronic messages are not limited to characters but can also include a range of *attachments*, including photographs, text, graphics, video, and sound. Finally, email is a *broadcast medium*. A computer can send a letter to a thousand recipients as easily as it can send it to one. (This feature has led to an explosive growth in **spam**—unsolicited bulk email. It is estimated that 70% of all email on the Internet today is spam.)

An interesting email-related application that flourished during the 1980s and 1990s was the **bulletin board system**, usually abbreviated BBS. A bulletin board is a shared public file where anyone can post messages and everyone is free to read the postings of others. It was an electronic version of the bulletin boards commonly seen in grocery stores, cafés, and public libraries. Today BBSs have evolved into **Internet forums** and **chat rooms** that support the real-time exchange of messages. Another popular form of message exchange is **texting** (also known as SMS, for Short Message Service) that allows for the rapid exchange of short messages, often using wireless cell phone technology. Today, roughly 100 billion text messages are sent on the Internet every day, and texting might be the latest killer app!

Main content

Chapter Contents

7.4.2 Social Networking

While some early email messages did contain technical content, even more were of the "Wanna meet for lunch today?" variety. Linking people together for purposes of social interaction has been a popular use of computer networks since their earliest days. Following the enormous success of email in the early 1970s, there were many other attempts to foster online communities. In 1980, a system called *Usenet* was developed. It was similar to a BBS with the added feature of having *newsgroups*—subgroups with a mutual interest in one specific topic, such as space flight, Chinese cooking, or Minnesota Vikings football. BBSs and Usenet were popular from the 1980s until the mid-1990s when they began to be replaced by web-based applications with a similar goal—allowing people to exchange thoughts, ideas, opinions, and stories. However, the web's powerful graphics capabilities and hypertext linking features allowed the scope of that sharing to increase dramatically, from text to all types of sound, graphics, videos, and imaging, and from the simple exchange of messages to advanced features such as mobile

access, online profiles, trust-based recommendation systems, "friending" controls, blog postings, contact lists, privacy management, and geosocial networking that organize users on the basis of geographic location. The applications that support these types of social exchanges are called **social networks**.

The use of web-based social networks has grown to the point where they are now some of the most well-known and widely used applications on the Internet, and there is hardly a person who is not thoroughly familiar with them and, more likely, a registered member of one or more. *Facebook*, developed in 2004 by Mark Zuckerberg while a student at Harvard, currently has 1.97 billion active users worldwide (about 24% of the Earth's population), and it receives over 1 billion visits per month. Other highly popular social media sites include YouTube (1 billion users), Wei'xin or WeChat in English (889 million users), QZone (595 million users), Instagram (600 million users), Tumblr (550 million users), and Twitter (319 million users).*

However, social networks are not without their issues, many of them serious and potentially dangerous. For example, the information you post on a social networking site can be used by people in ways that you never intended. Strangers may use the information contained in a profile to violate your privacy. Those who want to harm you could post false information to ruin your reputation or limit your access to high-quality education and employment. We will discuss some of the ethical and legal issues related to social networking in Chapter 17.

Main content

Chapter Contents

7.4.3 Resource Sharing

Another important network service is **resource sharing**, the ability to share *physical resources*, such as a 3-D printer or massive terabyte storage device, as well as *logical resources*, such as software and data, among multiple users.

The prices of computers and peripherals have been dropping for many years, so it is tempting to think that everyone can buy their own specialized I/O units or mass storage devices. However, that is not always a cost-effective way to configure computer systems. For example, a high-volume color laser printer may be used infrequently. Buying everyone in the office his or her own printer would leave most of the printers idle for long periods of time and would cost huge amounts for all the toner cartridges. It would be far more efficient to have a few shared printers, called **print servers**, which can be accessed whenever needed. Similarly, if a group of users requires access to a data file or a piece of software, it may make sense to keep a single copy on a set of shared network disks, called a **file server**. A network file server can be a cost-effective way to provide backup services as well as make it easier to access information from multiple devices such as your desktop computer at work, your tablet computer at home, and your smartphone on the road.

The style of computing wherein some nodes provide shared services while the remaining nodes are users (or clients) of those services is called, naturally enough, **client/server computing**. We have named two examples—print servers and file servers—but there are many others, such as mail servers, compute servers, and web

servers. The philosophy behind the **client/server model** is that we use a network to share resources that are too widespread, too expensive, or used too infrequently to warrant replication at every node. A diagram of the client/server model of computing is shown in Figure 7.24. (We discuss the next generation of client/server computing, called the cloud computing model, in Section 7.5)

Information sharing is another important service, and a network is an excellent way to access scientific, medical, legal, and commercial data files stored on systems all over the world. (In fact, it was the need to share information efficiently among hundreds of physicists that led to the development of the World Wide Web in the early 1990s.) For example, information can be distributed among the geographically dispersed sites of a multinational corporation and shared as needed, using a **distributed database**. Webpages can be exchanged between remote systems. Files can be transmitted anywhere in the world using FTP, which is mentioned in Figure 7.22.

Figure 7.24 The client/server model of computing

Many network sites now provide a service called a **data warehouse**. These nodes integrate massive amounts of information from a number of disparate sources and allow it to be electronically searched for specific facts or documents. Frequently, these sites focus on a single specialized topic, such as geopolitical data, stock price trends, real estate records, movie trivia, or information on case law and legal precedents. These single-subject data warehouses are often called **data marts**. Today, it is far more common for students, scientists, business people, and politicians to search for information in data marts than in the stacks of a physical library.

Another important resource-sharing service is the ability to support collaborative group efforts in producing a shared document such as a user's manual, grant application, or design specification. Workers on a project can communicate via the network; hold virtual conferences; check electronic calendars and schedule meetings automatically; and share, discuss, and edit documents online. A rapidly growing network application is *collaborative software*, also known as **groupware**. This is software that facilitates the creative efforts of individuals connected by a network and working on a single, shared project. One popular form of groupware is called a **wiki** (the Hawaiian word for fast or quick), which is a set of webpages that everyone is free to access, add to, or modify. One of the most successful examples of a wiki is *Wikipedia*, an online encyclopedia written, reviewed, and maintained by about 76,000 active volunteers working independently. Wikipedia currently includes 5.4 million English-language articles as well about 39 million other entries in 285 different languages from Polish to Portuguese, French to Finnish, Tajik to Tagalog. (By comparison, the *Encyclopedia Britannica* has about 0.5 million articles.) Currently, just the English pages in Wikipedia are viewed at the rate of 5.8 million per hour.

7.4.4 Electronic Commerce

Electronic commerce (or just **ecommerce**) is a general term applied to any use of computers and networking to support the paperless exchange of goods, information, and services in the commercial sector. The idea of using computers and networks to do business has been around for some time; the early applications of ecommerce include

- (1)

  the automatic deposit of paychecks,

- (2)

  automatic teller machines (ATMs) for handling financial transactions from remote sites, and

- (3)

  the use of scanning devices at checkout counters to capture sales and inventory information in machine-readable form.

More recently, the focus has been on the use of the Internet and the web to advertise and sell goods and services. Initially, the Internet was used mostly by scientists and engineers. However, the business world soon came to appreciate the potential of a communications medium that could cheaply and reliably reach millions of people around the world. In the last 10–15 years, traffic on the Internet has changed from primarily academic and professional to heavily commercial.

Today, about 8% of retail sales in the United States is handled online, up from less than 1% in 2001. This amounts to about $335 billion annually. Worldwide, online retail sales were $1.25 trillion in 2013 and growing at an annual rate of about 18%. Furthermore, using sites such as eBay, Amazon, and Craigslist, individuals, not just corporations, can reach a national or even international audience to buy and sell their products.

We will have much more to say about ecommerce and commercial uses of the Internet in Chapter 14.

Main content

Chapter Contents

7.5 Cloud Computing

In Section 7.4.3 we outlined reasons for the development of the client/server model shown in Figure 7.24. It is not cost effective to replicate expensive and infrequently used hardware, software, or data resources on every machine on your campus, office, or research center. Because of the economies of scale, it is far cheaper for an organization to purchase a shared resource (such as a server) and make it available on demand to users (clients) via a local area network. For over two decades the client/server model was the most widely used technique for sharing computational resources.

However, this model is not without problems, many of them quite serious. For example, the client/server approach can require large up-front capital expenditures to purchase the server, buy the software needed to access it, create the appropriate physical space,

and install the system. The new server may incur significant operating costs for network connections, power, cooling, spare parts, staff training, documentation, and repair. When clients request enhanced services, your organization either has to purchase the necessary upgrades or design, implement, and test these new services using in-house technical staff. Finally, you must purchase enough capacity to handle the maximum theoretical needs of the entire client community. Otherwise, when a user requests a service he or she could be turned down because the system is overloaded. Unfortunately, providing sufficient capacity for peak needs leaves servers underutilized a majority of the time.

Because of these shortcomings, a new model for shared access to computing resources has begun to emerge, a model called **cloud computing**. (The term comes from the cloud diagram used to represent this model, as shown in Figure 7.25.) Cloud computing behaves much like the client/server model of Figure 7.24 in that there are server nodes that provide services and client nodes that access those services. However, with cloud computing the servers no longer need to be local to the client population, and they no longer must be provided by your own organization.

Figure 7.25 Model of cloud computing

A client requests the services of a server via a communication network such as the Internet using a desktop computer, laptop, tablet, or mobile device, but the client has no idea of where the physical server is or even if the server is a single device or part of a larger **server farm**—an integrated collection of machines providing services over a network that would not be possible using only a single device. The logical concept of a computational resource (hardware, software, data) has been divorced from the physical realization of how that service is provided (its location, manufacturer, ownership, or technical structure). The term for the separation of a service from the entity (or entities) providing that service is **virtualization**, and it is one of the fundamental properties of cloud computing. For example, you use the concept of virtualization whenever you back up a smartphone or tablet to "the cloud." You have no idea where this data is going, how it is being stored, or who is managing the storage. You don't even know if it is being stored in a single location or distributed to multiple machines around the globe. You only know that it has been stored securely, and that you will be able to retrieve it if and when it is needed. Contrast this approach with how you backed up the hard drive on your desktop just a few years back. You would copy the information to a flash drive or an external hard drive that you purchased for this task and for which you took full responsibility to keep safe and secure.

The philosophy of cloud computing is to sell hardware, software, and data capabilities as complete, prepackaged services in which the technical details of networking, management, support, and implementation are hidden from users, who simply request and receive designated services at a specified cost. The user is freed from the messy and often quite complex details about how that service is being provided.

There are many types of cloud services provided by a huge number of cloud computing companies, but the services typically fall into one of three categories. The simplest and most basic is *infrastructure services*, where a provider offers access to shared resources

that users do not wish to purchase and maintain themselves, resources such as storage and data backup facilities or access to specialized input/output devices. This is the type of service you use whenever you back up a tablet or smartphone to the cloud rather than to your own desktop or laptop. *Application services* provide shared access to software packages such as email, games, accounting, finance, or document creation. By sharing these software resources users are freed from the costs associated with buying, maintaining, and upgrading these expensive packages. This is the type of cloud service you exploit if you use Google Mail to manage your email or Google Drive to collaborate with others in writing and editing shared documents. The most sophisticated form of cloud computing is *platform and development services*. Companies that create new software such as apps for the Apple iPhone often need sophisticated and expensive development platforms to design, code, and test their new applications. Cloud computing can provide, in a totally transparent way, a powerful software development environment that includes such tools as language translators, debuggers, testers, efficiency metrics, and documentation systems—basically everything a programmer could possibly need to create and market new software packages.

The cloud computing approach to shared resources in Figure 7.25 offers a number of advantages over the client/server model in Figure 7.24. It has the potential to ***lower costs*** since users pay for only what they need and what they use, not for what they have to buy. Cloud computing providers argue that by purchasing computing and communication services, rather than installing them on site, users are able to focus on their core mission (e.g., education, health, research) rather than on purchasing, maintaining, and upgrading expensive computing infrastructures. Cloud computing provides ***elasticity of demand*** as your needs fluctuate. Since implementation details of the server are hidden, a service provider is free to dynamically allocate more (or fewer) resources to match current needs. An organization no longer has to purchase sufficient capacity to meet peak client needs, a situation that may occur infrequently. Cloud computing also can offer ***wider access to shared resources***. Since a server is no longer tethered to your private local area network, it can typically be accessed anywhere in the world over the Internet using a smartphone, tablet, or laptop. Finally, some advanced computing and communications services might simply be too expensive for an organization to provide for themselves. Enhanced capabilities such as automatic backups and data replication (offering increased ***reliability***) or advanced firewalls and data encryption (offering improved ***security***) are just two examples of the many services available from cloud computing providers.

In the past two chapters we have described how system software can create a virtual environment that is easier for users to operate and understand. In the last 50 or so years, that virtual environment has grown ever more powerful and provided users with greater access to an enormous range of resources. The next step in that ongoing process is cloud computing. With cloud computing, not only are the resources of a single system and shared local servers accessible in a transparent way, the entire communications network as well as a global network of computational facilities becomes available in a totally transparent manner. With the simple push of a key, click of a mouse, tap of a finger, or voice command, a plethora of powerful services is available to a user without any concern about who is providing this service or where it is happening. This is certainly a most powerful virtual environment in which to work.

7.6 A History of the Internet and the World Wide Web

In the preceding sections, we discussed the technical characteristics and services of networks in general. However, to most people, the phrase *computer network* isn't a generalized term but a very specific one—the global Internet and its most popular component, the World Wide Web.

In this section, we highlight the history, development, and growth of the Internet and the World Wide Web. Much of the information in the following pages is taken from the original 1997 article "A Brief History of the Internet," written by its original designers and available on the web.*

7.6.1 The Internet

The Internet is an idea that has been floating around for more than 50 years. The concept first took shape during the early 1960s and was based on the work of computer scientists at MIT and the RAND Corporation in the United States and the NPL Research Laboratory in Great Britain. The first proposal for building a computer network was made by J. C. R. Licklider of MIT in August 1962. He wrote his colleagues a memo titled (somewhat dramatically) "The Galactic Network," in which he described a globally interconnected set of computers through which everyone could access data and software. He convinced other researchers at MIT, including Larry Roberts and Leonard Kleinrock, of the validity of his ideas. From 1962 to 1967, they and others investigated the theoretical foundations of wide area networking, especially such fundamental technical concepts as protocols, packet switching, and routing.

In 1966, Roberts moved to the Advanced Research Projects Agency (ARPA), a small research office of the U.S. Department of Defense charged with developing technology that could be of use to the U.S. military. ARPA was interested in packet-switched networking because it seemed to be a more secure form of communications during wartime. ARPA funded a number of network-related research projects, and in 1967 Roberts presented the first research paper describing ARPA's plans to build a wide area packetswitched computer network. For the next two years, work proceeded on the design of the network hardware and software. The first two nodes of this new network, called the ARPANET, were constructed at UCLA and the Stanford Research Institute (SRI), and in October 1969, the first computer-to-computer network message was sent. (The very first Internet transmission was sent by a UCLA student programmer named Charles Kline, and it contained the single word *login*. The network then crashed!) Later that same year two more nodes were added (the University of California-Santa Barbara

and the University of Utah), and by the end of 1969, the budding four-node network was off the ground.

The ARPANET grew quickly during the early 1970s, and it was formally demonstrated to the scientific community at an international conference in 1972. It was also in late 1972 that the first killer app was developed—electronic mail. It was an immediate success and caused an explosion of growth in people-to-people traffic rather than the people-to-machine or machine-tomachine traffic that dominated usage in the first few years.

The success of the ARPANET in the 1970s led other researchers to develop similar types of computer networks to support information exchange within their own scientific area: HEPNet (High Energy Physics Network), CSNET (Computer Science Network), and MFENet (Magnetic Fusion Energy Network). Furthermore, corporations started to notice the success of the ARPANET and began developing proprietary networks to market to their customers: SNA (Systems Network Architecture) at IBM and DECNet from Digital Equipment Corporation. The 1970s were a time of rapid expansion of networks in both the academic and the commercial communities.

Farsighted researchers at ARPA, in particular Robert Kahn, realized that this rapid and unplanned proliferation of independent networks would lead to incompatibilities and prevent users on different networks from communicating with each other, a situation that brings to mind the problems that national railway systems have sharing railcars because of their use of different gauge track. Kahn knew that to obtain maximum benefits from this new technology, all networks need to communicate in a standardized fashion. He developed the concept of **internetworking**, which stated that any WAN is free to do whatever it wants internally. However, at the point where two networks meet, both must use a common addressing scheme and identical protocols—that is, they must speak the same language.

This is the same concept that governs the international telephone system. Every country is free to build its own internal phone system in whatever way it wants, but all must agree to use a standardized worldwide numbering system (country code, city code), and each must agree to send and receive telephone calls outside its borders in the format standardized by the worldwide telephone regulatory agency.

Figure 7.26 is a diagram of a "network of networks." It shows four WANs called A, B, C, and D interconnected by a device called a **gateway** that makes the internetwork connections and provides routing between different WANs.

Figure 7.26A network of networks

To allow the four WANs of Figure 7.26 to communicate, Kahn and his colleagues needed to create

- (1)

  a standardized way for a node in one WAN to identify a node located in a different WAN and
- (2)

a universally recognized message format for exchanging information across WAN boundaries.

Kahn, along with Vinton Cerf of Stanford, began working on these problems in 1973, and together they designed the solutions that became the framework for the Internet. Specifically, they created both the hierarchical host naming scheme that we use today and the TCP/IP protocols that are the "common language" spoken by networks around the world. (These protocols were discussed in Sections 7.3.3 and 7.3.4.)

During the late 1970s and early 1980s, work proceeded on implementing and installing TCP/IP on not only mainframe computers but also the PCs and desktop machines that were starting to appear in the marketplace. It is a tribute to the power and flexibility of the TCP/IP protocols that they were able to adapt to a computing environment quite different from the one that existed when they were first created. Originally designed to work with the large mainframe computers of the 1970s, they were successfully implemented in a new computing environment—desktop PCs connected by LANs. (Later they were again successfully moved to the computing environments of the 21st century—tablets, smartphones, and wireless communications.)

By the early 1980s, TCP/IP was being used all around the world. Even networks that internally used other communication protocols implemented TCP/IP to exchange information with nodes outside their own community. With TCP/IP becoming a *de facto* networking standard, a global addressing scheme, and a growing set of important applications, the infrastructure was in place for the creation of a truly international network. By the early 1980s the Internet, in its modern form, had slowly begun to emerge.

Although many of the technical problems had been solved, networking had yet to make a significant impact on the general population for one important reason: To access the ARPANET you needed a research grant from the U.S. Department of Defense (DOD). In the early 1980s, the people using the Internet were almost exclusively physicists, engineers, and computer scientists at a select set of secure military bases and academic research centers. For example, in 1982, 13 years after its creation, there were only 235 computers connected to the ARPANET.

One last step was needed, and it was taken by the National Science Foundation (NSF) in 1984. In that year, the NSF initiated a project whose goal was to bring the advantages of the Internet to the *entire* academic and professional community, regardless of discipline or relationship with the DOD. The NSF planned and built a national network called *NSFNet*, which used TCP/IP technology identical to the ARPANET. This new network interconnected six NSF supercomputer centers with dozens of new regional networks set up by the NSF. These new regional networks included thousands of users at places like universities, government agencies, libraries, museums, medical centers, and even high schools. Thus, by the mid-1980s, this emerging "network of networks" had grown to include many new sites and, even more important, a huge group of first-time users, such as students, faculty, librarians, museum staff, politicians, civil servants, and urban planners, to name just a few.

At about the same time, other countries began developing wide area TCP/ IP backbone networks like NSFNet to interconnect their own universities, research centers, and government agencies. As these national networks were created, they were linked into this expanding network, and the user population continued to expand. For the first time

since the development of networking, the technology had begun to have an impact on the wider community. A diagram of the state of internetworking in the late 1980s is shown in Figure 7.27.

Figure 7.27 State of networking in the late 1980s

Some time in the late 1980s, the term ARPANET ceased to be used because, as Figure 7.27 shows, the ARPANET was now only one of many networks belonging to a much larger collection. (By 1990, the collection had grown to 300,000 computers on 3,000 separate networks.) People began referring to the entire set of interconnected networks as "the Internet," though this name was not officially adopted by the U.S. government until October 24, 1995.

Once the public had easy access, the Internet became an immediate success and grew rapidly. By the middle of 1993, it included 1.8 million host computers and roughly 5 to 10 million active users, and its size was doubling every year. In fact, it had become so successful that the NSF decided it was time to get out of the "networking business." The goal of the NSF is to fund basic research, not to operate an ongoing commercial enterprise. In April 1995, NSFNet closed up shop. The exit of the U.S. government from the networking arena created business opportunities for new firms called Internet service providers (ISPs) that offered the Internet access once provided by ARPANET and NSFNet.

By early 2017, the Internet had grown to more than 1.06 billion computers and 3–4 billion users located in just about every country in the world, and the extraordinary growth of the Internet continues to this day. Figure 7.13 in Section 7.2.4 shows a graph of the number of host computers connected to the Internet.

The Internet has been one of the biggest success stories in moving research out of the laboratory and into the wider community. What began as the wild idea of a few dedicated researchers has grown, in only 50 years, into a global communications infrastructure moving trillions of bits of data among billions of people. It has adapted time and time again—to changes in usage (from research and academic to commercial and entertainment), changes in hardware (from mainframes to laptops, tablets, and smartphones), and changes in scale (from hundreds of nodes to billions of nodes).

The Internet continues to undergo massive growth and change, this time from the most important new killer app developed for the Internet since email—the World Wide Web.

Main content

Chapter Contents

7.6.2 The World Wide Web

Tim Berners-Lee, a researcher at CERN, the European High Energy Physics Laboratory in Geneva, Switzerland, first developed the idea for a hypertext-based information distribution system in 1989. Because physics research is often done by teams of people

from different universities, he wanted to create a way for scientists throughout Europe and North America to easily exchange information such as research articles, journals, and experimental data. Although they could use existing Internet services such as FTP and email, Berners-Lee wanted to make information sharing easier and more intuitive for people unfamiliar with computer networks.

Beginning in 1990, Berners-Lee designed and built a system using the concept of **hypertext**, a collection of documents interconnected by pointers, called *links*, as shown in Figure 7.28. Traditional documents are meant to be read linearly from beginning to end, but users of hypertext documents (called *pages* in web parlance) are free to navigate the collection in whatever order they want, using the links to move freely from page to page. Berners-Lee reasoned that the idea of hypertext matched up very well with the concept of networking and the Internet. Hypertext documents could be stored on the machines of the Internet, and a link would be the name of a page along with the IP address of the machine where that page is stored. He called his hypertext link a URL, an acronym for Uniform Resource Locator, and it is the worldwide identification of a webpage located on a specific host computer on the Internet.

Figure 7.28 Hypertext documents

Geography Lesson

The Internet is a truly "global phenomenon," affecting the way people work, shop, and communicate throughout the world. Consider that, whereas the United Nations has 192 member states, the Domain Name System (DNS) of the Internet includes entries for 239 countries, territories, and possessions. The DNS includes standardized domain names for such places as (you may want to get out your atlas) Comoros (.km), Nauru (.nr), Pitcairn (.pn), Kiribati (.ki), Mayotte (.yt), Guinea Bissau (.gw), and St. Pierre and Miquelon (.pm). As of 2017, the smallest DNS domains are .td, the Republic of Chad and .sj, Svalbard and Jan Mayen Islands. They both contain exactly zero host computers!

Berners-Lee named his new information system the **World Wide Web**, and it was completed and made available to all researchers at CERN in August 1991, the date that marks the birth of the web. It became an instant success, and traffic on the CERN web server increased by 1,000% in its first two years of use. In April 1993, the directors of CERN, realizing the beneficial impact that the web could have on research throughout the world, announced that, effective immediately, all web technology developed at CERN would be freely available to everyone without fees or royalties. For many people, this important announcement really marks the emergence of the World Wide Web on a global scale.

A powerful graphical web browser, called Mosaic, was developed in late 1993 and made available to the general public so that they could begin to use this new service. With the appearance of Mosaic, the World Wide Web began to take off. It was a network application that offered users exactly what they needed most—easy access to massive amounts of helpful information whenever they wanted it. Other browsers soon appeared in the marketplace, including Netscape Navigator (1994), Microsoft Internet Explorer (1995), Apple Safari (2003), Mozilla Firefox (2004), Google Chrome (2008), and Microsoft Edge (2015).

In late 1995, the NSF conducted a study of the different types of traffic on the Internet as a percentage of all information sent. At that time, the World Wide Web represented 23.9% of the total volume of Internet traffic, even though it had been in existence for only four years!

Since that time, the web has continued to grow exponentially, containing roughly 1.2 billion active websites by 2017. It is by far the fastest growing component of the Internet. (Although today only about 38% of all web traffic originates from living human beings! The other 62% of traffic comes from *bots*, software applications designed to carry out automated tasks such as indexing webpages.) The web's colorful graphics and simple point-and-click method of accessing information has made it the most important method of delivering the capabilities of networking to everyone—from toddlers to senior citizens and kindergarten students to PhDs. For many people, the web *is* the Internet.

Net Neutrality

Imagine you are building a new dream home and have connected it to your city's public water system. Having made this connection you turn on the tap only to get an agonizingly slow drip, while your next door neighbor has a powerful flow from every outlet. What could possibly cause such a disparity in service?

What if you learned that your neighbor had paid a bit more each month to get better water service, even though it could negatively impact the amount of water available to others? You would be furious because the water company is a *public* utility that is supposed to provide equal levels of service to all customers.

That is exactly the argument raging right now with regard to the Internet. The term **net neutrality** means that a public information network such as an Internet service provider (ISP) should treat all users, all platforms, and all content equally—essentially it should be operated like the water, electric, and gas companies, as publicly regulated utilities. The proponents of this concept argue that if net neutrality were the law of the land, all users would be able to exchange information freely and openly without worrying about regulations, restrictions, and controls imposed on them by a third party such as their ISP. However, opponents of net neutrality say that providing a tiered level of Internet service is no different than creating high-speed lanes on a public roadway where people pay tolls to avoid heavy traffic and get to their destination more quickly. Essentially opponents of net neutrality want to create a "fast lane" on the Internet to provide top-level service for their best customers.

The Federal Communications Commission (FCC) held numerous hearings on the advantages and disadvantages of net neutrality and whether or not to permit paid prioritization of communication services. In April 2017 the Chairman of the FCC put forward a proposal to terminate all federal regulations regarding net neutrality that had been in place since 2014. This plan created a good deal of disagreement and unhappiness, and only time will tell what will eventually happen.

Main content

Chapter Contents

7.7 Conclusion

Computer networking has changed enormously in the 50 years that it has been around. From a specialized communication system devoted to academic research, it has

blossomed into a worldwide information system. What was once the esoteric domain of a few thousand scientists is now used by billions of people, the vast majority of whom have no formal training in computer science. From providing access to technical databases and research journals, it has become a way for the average citizen to shop, chat, stay informed, and be entertained. There is every reason to believe that the Internet will continue to grow and evolve as much in the coming years as it has in the past.

The most pressing issue facing the Internet today is not better technology or new killer apps. Those issues have been and will continue to be addressed and solved. The biggest concern today is how the growth and direction of networking will be managed and controlled. In its early days, the Internet was run by a core group of specialists without a financial stake in its future, and its management was relatively simple. Now that it is a global phenomenon that affects billions of people and generates hundreds of billions of dollars in revenue, the Internet is being pulled and tugged by many new constituencies and stakeholders, such as corporations, politicians, lawyers, advertisers, government agencies, and manufacturers. The important question now is who will speak for the Internet in the future and who will help shape its destiny. As the designers of the Internet warned at the end of their paper (see footnote 4) on the history of networking:

If the Internet stumbles, it will not be because we lack for technology, vision, or motivation. It will be because we cannot set a direction and march collectively into the future.