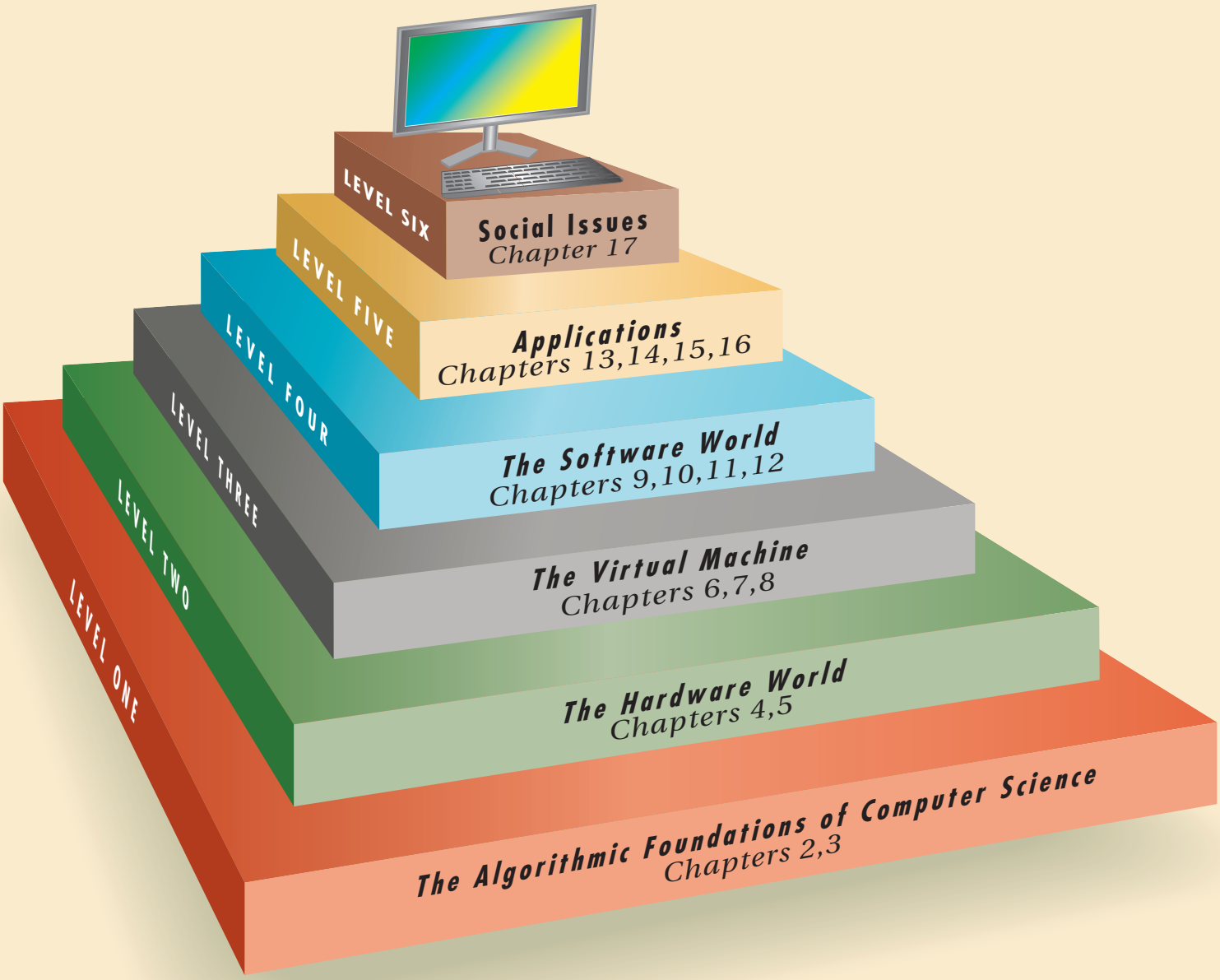

FIFTH EDITION
INVITATION TO
COMPUTER SCIENCE
G. Michael Schneider • Judith L. Gersting

5TH EDITION

Invitation to Computer Science


LEVEL SIX
Social Issues
Chapter 17
LEVEL FIVE
Applications
Chapters 13,14,15,16
LEVEL FOUR
The Software World
Chapters 9,10,11,12
LEVEL THREE
The Virtual Machine
Chapters 6,7,8
LEVEL TWO
The Hardware World
Chapters 4,5
LEVEL ONE
The Algorithmic Foundations of Computer Science
Chapters 2,3

5TH EDITION

Invitation to Computer Science

G. Michael Schneider
Macalester College

Judith L. Gersting
University of Hawaii, Hilo

Contributing author:
Keith Miller
University of Illinois, Springfield

COURSE TECHNOLOGY
CENGAGE Learning

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

Invitation to Computer Science, Fifth Edition
G. Michael Schneider and Judith L. Gersting

Executive Editor: Marie Lee

Acquisitions Editor: Amy Jollymore

Senior Product Manager: Alyssa Pratt

Development Editor: Deb Kaufmann

Editorial Assistant: Julia Leroux-Lindsey

Marketing Manager: Bryant Chrzan

Content Project Manager: Jennifer K. Feltri

Art Director: Faith Brosnan

Cover Designer: RHDG/Tim Heraldo

Cover Artwork: Fotolia.com (Royalty Free), Image # 375162

Compositor: Integra

ISBN-13: 978-0-324-78859-4
ISBN-10: 0-324-78859-2

Course Technology
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:
international.cengage.com/region

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For your lifelong learning solutions, visit **course.cengage.com**
Visit our corporate website at **cengage.com**.

Printed in Canada
1 2 3 4 5 6 7 13 12 11 10 09

BRIEF CONTENTS

Chapter 1 An Introduction to Computer Science 1

LEVEL 1 The Algorithmic Foundations of Computer Science 36

Chapter 2 Algorithm Discovery and Design 39
Chapter 3 The Efficiency of Algorithms 79

LEVEL 2 The Hardware World 126

Chapter 4 The Building Blocks: Binary Numbers, Boolean Logic, and Gates 129
Chapter 5 Computer Systems Organization 187

LEVEL 3 The Virtual Machine 236

Chapter 6 An Introduction to System Software and Virtual Machines 239
Chapter 7 Computer Networks, the Internet, and the World Wide Web 287
Chapter 8 Information Security 333

LEVEL 4 The Software World 356

Chapter 9 Introduction to High-Level Language Programming 359
Chapter 10 The Tower of Babel 397
Chapter 11 Compilers and Language Translation 445
Chapter 12 Models of Computation 491

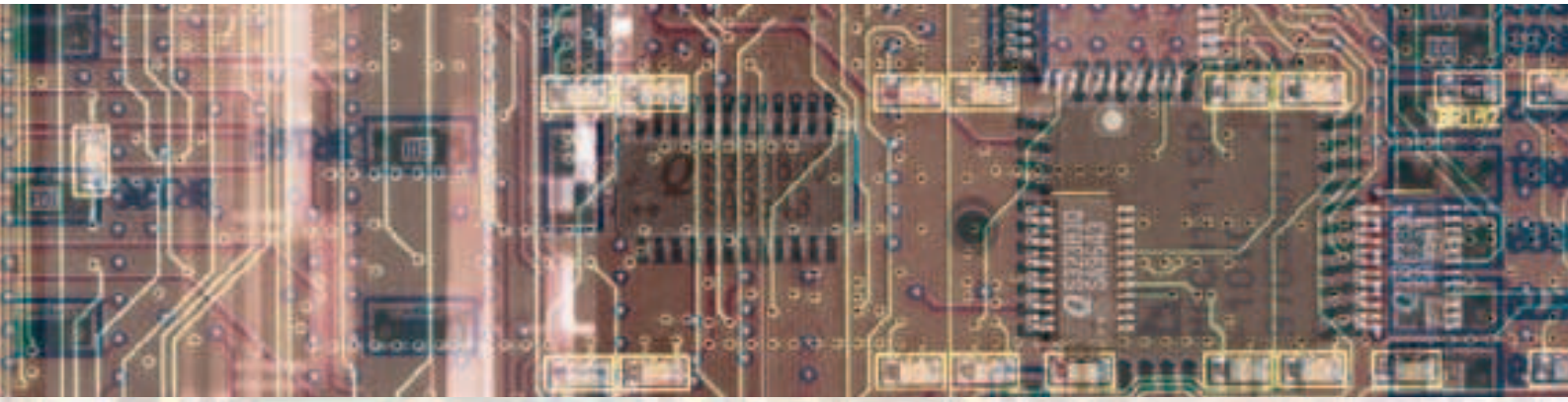
LEVEL 5 Applications 532

Chapter 13 Simulation and Modeling 535
Chapter 14 Electronic Commerce and Databases 561
Chapter 15 Artificial Intelligence 585
Chapter 16 Computer Graphics and Entertainment: Movies, Games, and Virtual Communities 617

LEVEL 6 Social Issues in Computing 642

Chapter 17 Making Decisions about Computers, Information, and Society 645

Answers to Practice Problems 673
Index 699



CHAPTER 7

Computer Networks, the Internet, and the World Wide Web

7.1 Introduction
7.2 Basic Networking Concepts
- **7.2.1** Communication Links
- **7.2.2** Local Area Networks
- **7.2.3** Wide Area Networks
- **7.2.4** Overall Structure of the Internet

7.3 Communication Protocols
- **7.3.1** Physical Layer
- **7.3.2** Data Link Layer
- **7.3.3** Network Layer
- **7.3.4** Transport Layer
- **7.3.5** Application Layer

LABORATORY EXPERIENCE 11

7.4 Network Services and Benefits
7.5 A Brief History of the Internet and the World Wide Web
- **7.5.1** The Internet
- **7.5.2** The World Wide Web

7.6 Conclusion
EXERCISES
CHALLENGE WORK
FOR FURTHER READING

7.1 Introduction

Every once in a while there occurs a technological innovation of such importance that it forever changes a society and the way its people live, work, and communicate. The invention of the printing press by Johannes Gutenberg in the mid-fifteenth 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 eighteenth and early nineteenth 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 our own century, we are certainly aware of the massive social changes, both good and bad, wrought by inventions like the telephone, automobile, television, and computer.

Many people feel that we are 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 exchanging resources and information. During the early stages of network development, the only information exchanged was text such as e-mail, database records, and technical papers. However, the material sent across a network today can be just about anything—television and radio signals, voice, graphics, handwriting, photographs, and movies, to name just a few. If information can be represented in the 0s and 1s of binary (as described in Section 4.2), it can be transmitted across a network.

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

Networking could also foster the growth of democracy and global understanding by providing unrestricted access to newspapers, magazines, radio, and television, as well as support the unfettered exchange of diverse and competing thoughts, ideas, and opinions. 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 wish to prevent the dissemination of opposing ideas often move quickly to restrict both Internet and Web access.

In Chapter 6 we saw how system software can create 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 integrated resources. In this chapter we take a detailed look at the technology of computer networks—what they are, how they work, and the benefits they can bring. We also examine the most widely used network, the Internet, and its most important application, the World Wide Web.

7.2 Basic Networking Concepts

A **computer network** is a set of independent computer systems connected by telecommunication links for the purpose of sharing information and resources. The individual computers on the network are referred to as **nodes**, **hosts**, or **end systems**, and they range from PDAs (personal digital assistants) and tiny laptops to the massively parallel supercomputers introduced in Chapter 5. In this section we describe some of the basic 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 40 years that networks have existed, telecommunications facilities have undergone enormous change.

Blogs

One of the most important Web applications is the "blog," a contraction of the term Web log. A blog is a Web-based publication consisting of virtually any periodic articles that its writer(s) wish to share with the general public. Sometimes it contains nothing more than a daily journal—what I did today. More commonly the articles are political, social, or cultural essays that reflect the opinions and biases of the blog author(s). Whereas some blogs are produced by a community of like-minded people sharing responsibility for writing and posting articles, the majority are simply the thoughts and feelings of individuals with a computer and the necessary "blogware"—software for editing, organizing, and publishing on the Web. (According to Technorati, a blog rating and tracking Web site, as of mid-2008 there were about 112 million blogs worldwide with about 175,000 new sites coming online daily!)

Our history is filled with stories of individual crusaders who published fiery newsletters supporting or decrying some government policy. For example, the *Federalist Papers* by Alexander Hamilton and James Madison were written in support of the proposed U.S. Constitution. *The Liberator* was a fervent anti-slavery newsletter published in Boston by William Lloyd Garrison, a Quaker abolitionist. However, there was a limit to the audience that these early crusaders could reach, set by the cost of printing and the time required to distribute these newsletters to readers. (At the peak of its influence, *The Liberator* had a circulation of fewer than 3,000.) The Web has changed all that. It costs virtually nothing to write and post your thoughts on a Web page, and if your ideas become widely discussed (perhaps by being mentioned on TV, radio, or in the newspaper) a blog might be accessed and read by millions of readers.

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 callee. This circuit lasts for the duration of the call, and when you hang up it is terminated.

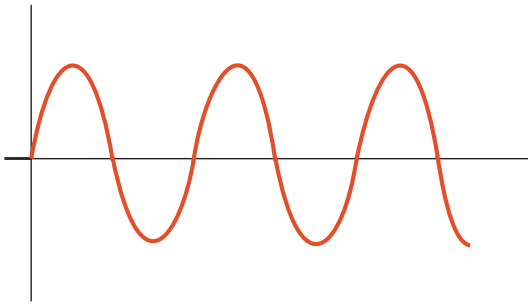
The voice-oriented dial-up telephone network was originally a totally **analog** medium. As we first explained in Chapter 4, this means that the physical quantity used to represent information, usually voltage level, is continuous and can take on any value. An example of this is shown in Figure 7.1(a). 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).

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** 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.

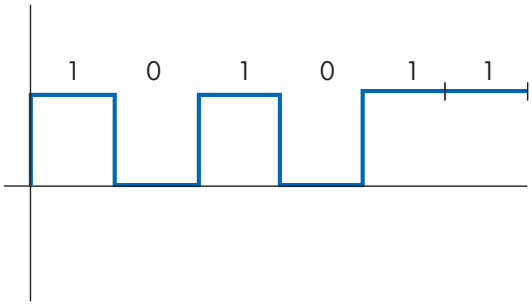
At the other end of the transmission line, a modem performs the inverse operation, which is called demodulation. (Modem is a contraction of the two terms *mo*dulation and *dem*odulation.) It takes the received waveform, separates the carrier from the encoded digital signal, and passes the digital data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

FIGURE 7.1

Two Forms of Information Representation



(a) Analog Representation



(b) Digital Representation

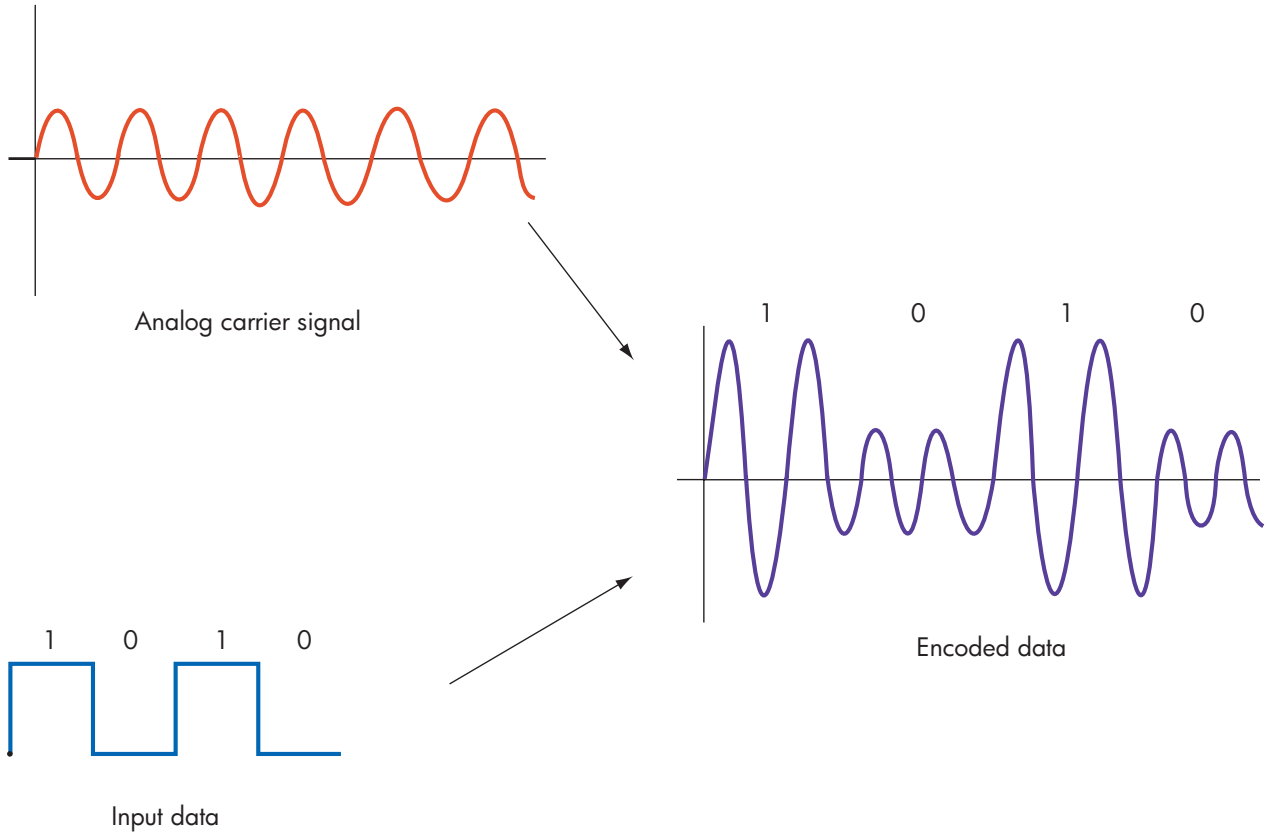


FIGURE 7.2

Modulation of a Carrier to Encode Binary Information

Initially, these analog encoding and decoding operations could not be done very quickly because of the high error rate and low capacity, or **bandwidth**, of a switched telephone line. In the early days of telecommunications—the 1970s and 1980s—the rate at which information could be sent and received via a phone line was limited to about 1,200–9,600 bits per second (bps). Advances in modem design have produced devices that now transmit at 56,000 bps, or 56 Kbps, an order-of-magnitude increase. However, this is still considered too slow to handle the transmission of multimedia-based documents such as Web pages, MP3 files, and streaming video.

The dial-up telephone system is still used occasionally for remote access to networks, and many computers are equipped with a built-in 56 Kbps modem. However, their limited speed makes dial-up phone links inconvenient for applications where speed is vital or we are sending large volumes of data.

A technology called **broadband** has rapidly been replacing modems and analog phone lines for data communications to and from our homes, schools, and offices. The term broadband generally refers to any communication link with a transmission rate exceeding 256,000 bps. In the case of home users, there are currently two widely available broadband options—digital subscriber line (DSL) and cable modem.

A **digital subscriber line** 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 their 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 Web page being simultaneously downloaded by someone else in the family.

Furthermore, unlike the modem which 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 Web page, 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 Web page you receive—complete with applets, graphics, and plug-ins—contains possibly millions of bits. To handle this imbalance, a DSL provides greater bandwidth coming in to your computer than going out. Typical DSL speeds are 2–8 million bits per second (Mbps) for downloads and 0.5–1 million bits per second for uploads—still much more than is available from a 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 download speeds roughly comparable to those available from DSL.

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 is too slow for some 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 in excess of 1 billion bits per second (Gbps). In the early 1990s, the U.S. government funded a long-term research project called NREN, the *National Research and Education Network*. One of its goals was to investigate the design and implementation of wide area gigabit data networks. The project was successful, and in 1998 the first international **gigabit Ethernet standard** was adopted by the **IEEE** (the **Institute of Electrical and Electronics Engineers**), an international professional society responsible for 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. Many classrooms and office buildings today are wired to support 10 Mbps, 100 Mbps, or even 1,000 Mbps—18,000 times faster than a 56K modem! In addition, most PCs today come 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 **ten-gigabit Ethernet** standard, a version of Ethernet with the almost unimaginable 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 entire contents of 1,700 books, each 300 pages long!

Do applications that truly need to transmit information at billions of bits per second exist? 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 a single movie frame, at different transmission speeds. As described in Section 4.2, a high-resolution color image contains at least 5 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 80,000,000 bits of data. If the image is compressed before it is sent, and the compression ratio is 10:1 (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 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. (You have probably had the experience of waiting for what seemed like forever as a Web page s-l-o-w-l-y appeared on your screen.) 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 a 10 Mbps Ethernet, and a blazing 0.08 second with 100 Mbps Ethernet.

However, even 0.08 second may not be fast enough if an application requires the rapid transmission of either 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 send 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 8,000,000 $\times$ 24 = 192 Mbps. This is beyond the speed of modems, DSL, cable modems, and even 100 Mbps Ethernet, but it is achievable using gigabit networks. These high-speed networks are widely used in such data-intensive applications as exchanging 3D medical images, transmitting weather satellite data, and supporting collaboration among researchers working on the Human Genome Project.

A relatively recent development in telecommunications is the growth of **wireless data communication** using radio, microwave, and infrared signals. In the wireless world, users no longer need to be physically connected to a wired network to access data, just as cellular phones liberated telephone users. Using wireless, you can be in the back yard, a car, at the beach, or on the factory floor and still send and receive e-mail, access online databases, or surf the Web. The ability to deliver data to users regardless of their location is called **mobile computing**.

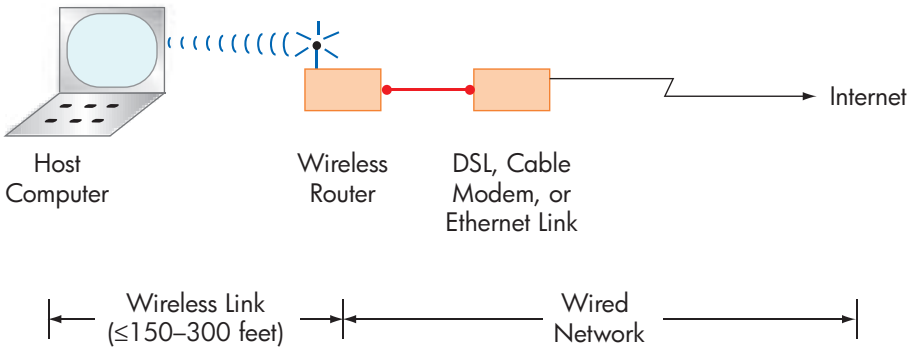
There are two forms of wireless data communications. In a **wireless local access network**, a user transmits from his or her computer to a local wireless **base station**, often referred to as a **wireless router**, that is no more than a few hundred feet away. This base station is connected to a traditional wired

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
Ten-gigabit Ethernet	10 Gbps	0.0008 second

network, such as a DSL or cable modem to provide full Internet access. This is the type of 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 the following diagram:



One of the most widely used standards for wireless local access is called **Wi-Fi**, short for **Wireless Fidelity**. It is also referred to by its official name, the **IEEE 802.11 wireless network standards**. 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 in stores and shops as a **Wi-Fi hot spot** (a router). Wi-Fi systems generally use the 2.4 GHz radio band for communications and support download transmission speeds of about 10-50 Mbps.

A new development in wireless networking is the concept of a **Metropolitan Wireless Local Access Network**. A number of cities in the U.S., Europe, and Asia have installed Wi-Fi routers every few blocks throughout the city, often on top of telephone poles or building roofs. These routers provide convenient, low cost wireless Internet access to all residents.

Another popular wireless local access standard is **Bluetooth**. It is a low-power wireless standard used to communicate between devices located quite close to each other, typically no more than 30–50 feet (10–15 meters). Bluetooth is often used to support communication between wireless computer peripherals, such as printers, mice, and keyboards and a laptop or desktop system close by. Bluetooth also supports information exchange between digital devices such as mobile phones, cameras, and video game consoles.

While Wi-Fi is great for communicating with a nearby router, its transmission limit means it cannot provide mobile Internet access from a car or outdoor site far from any base station. To handle this type of wireless communications we need a different type of network called a **wireless wide-area access network**. In this type of network the computer (typically a PDA or smart phone) 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 **3G**. It offers voice services as well as data communication at rates of 0.5 to 2.4 Mbps, comparable to DSL or cable modem. Future plans call for a speed increase to about 5-15 Mbps.

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. 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. While a few random "clicks" and "pops" do not disrupt voice communications over a mobile phone, it can be disastrous in data communications. For example, if you are transmitting data at 10 million bits per second, a break-up on the line that lasts only one-tenth of a second could potentially cause the loss of one million bits of data. Wireless is often slower than wired communication (a few Mbps rather than hundreds of Mbps or Gbps), which may make it inappropriate for the transfer of large amounts of data. Finally, there is the issue of security. Currently, it is not difficult to intercept transmissions and gain unauthorized access to wireless networks. All of these are ongoing concerns being investigated by the computer science and telecommunications research community.

PRACTICE PROBLEMS

1. Show how the 4-bit digital value 0110 is converted to an analog signal by a modem that modulated the *frequency* of a carrier wave, rather than its amplitude.
2. Consider an uncompressed 1,200 × 780 image, with each pixel stored using an 8-bit gray scale representation. If we want to transmit the entire image in under 1 second, what is the minimum acceptable transmission speed?

Ubiquitous Computing

The rapid growth of wireless communications, along with the availability of extremely cheap microprocessors, has led to an exciting new area of computer science research called **ubiquitous computing**, also called **pervasive computing**. In the early days of computing, a single large mainframe served many users. In the PC era, a single desktop machine served a single user. In the ubiquitous computing model, many computers work together to serve a single user, and rather than being perched on a desktop, they become nearly invisible. The idea is that computers will become so commonplace that they will blend into the background and disappear from our consciousness, much as electricity has today. The goal is to create a system that is embedded in the environment, providing its service in a seamless, efficient manner.

Computers will be located inside our appliances, furnaces, lights, clocks, and even clothing to provide useful services in a transparent fashion. Topics of research in this area include such things as **wearable computing** and **smart homes**. As described by Mark Weiser of Xerox, "Ubiquitous computing is invisible, *everywhere* computing that does not sit on the desktop but lies deep inside the woodwork."

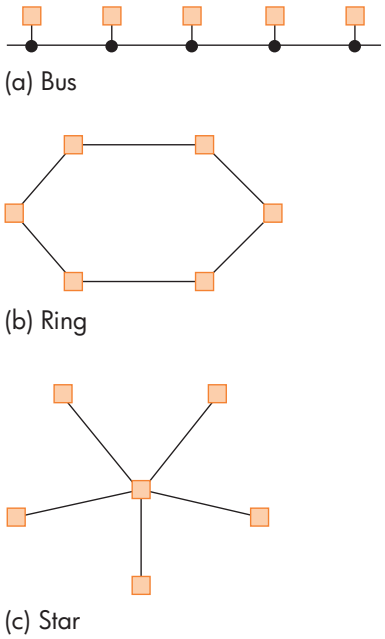
7.2.2 *Local Area Networks*

There are two types of computer networks. A **local area network (LAN)** connects hardware devices such as computers, printers, and storage devices that are all in close proximity. (A diagram of a LAN was provided in Figure 6.21.) Examples of LANs include the interconnection of machines in one room, in the same office building, or on a single campus. 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.

The previous section described how a wireless local network is 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 interconnection strategies, as seen in Figure 7.4. In the **bus** topology, Figure 7.4(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 Web page at the exact same time, then the effective transmission rate is lower than expected, because one of them must wait.

The **ring** topology of Figure 7.4(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** network, Figure 7.4(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.

FIGURE 7.4
Some Common LAN Topologies



There are many different LAN technologies available in the marketplace, but the most widely used is Ethernet, which you learned about in the previous section. It is the model that we will use to describe the general behavior of all LANs.

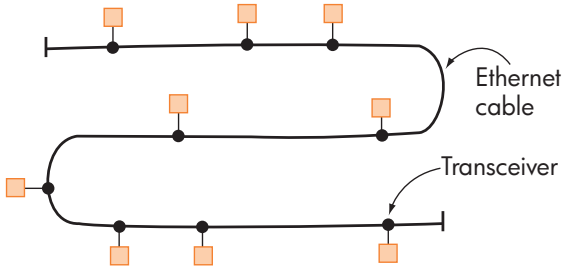
Ethernet uses the bus topology of Figure 7.4(a). To send a message, a node places the message, including the destination address, on the cable. Because the line is shared, the message is received by every other node (assuming no one else sent at the exact same time and garbled our data). Each node looks 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 7.5(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.

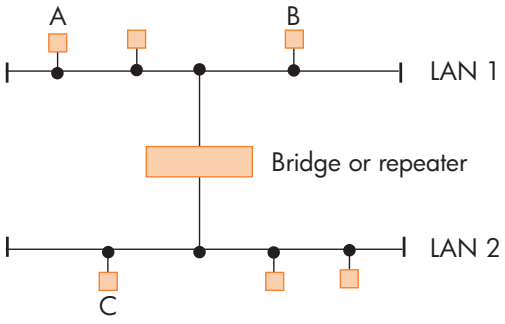
A **repeater** is a device that simply amplifies and forwards a signal. In Figure 7.5(b), if the device connecting the two LANs is a repeater, then every message on LAN1 is forwarded to LAN2, and vice versa. Thus, when two Ethernet LANs are connected by a repeater, they function exactly as if they were a single network.

A **bridge**, also called a **switch**, 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.

FIGURE 7.5
An Ethernet LAN Implemented Using Shared Cables



(a) Single Cable Configuration



(b) Multiple Cable Configuration

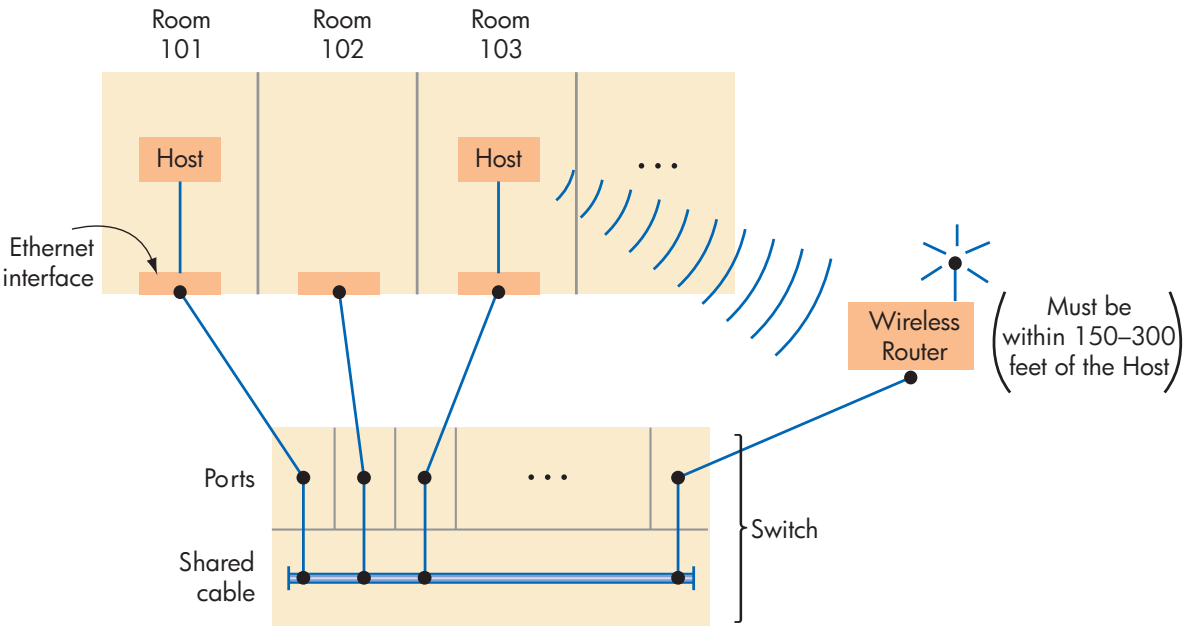


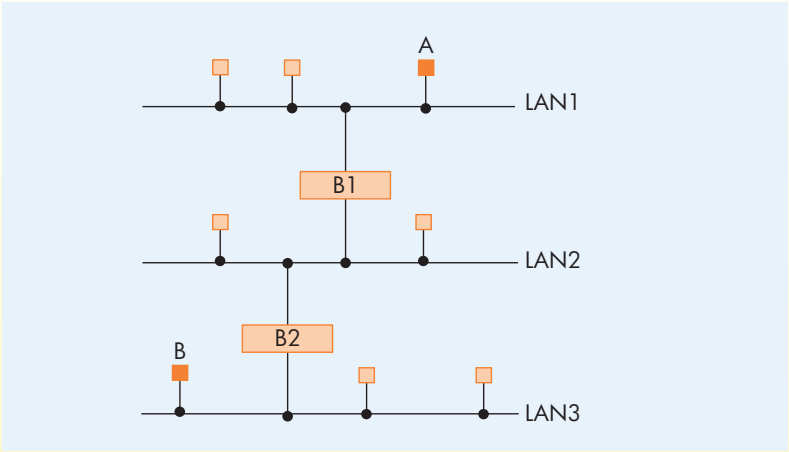
FIGURE 7.6
An Ethernet LAN Implemented Using a Switch

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, we first activate that port, typically by flipping a switch, and then simply plug our machine directly into the wall socket. This approach is shown in Room 101 of Figure 7.6. Alternately, we could use Wi-Fi to transmit from our computer to a wireless router located somewhere in the building. This router would then connect to one of the Ethernet ports in the hub. This approach is shown in Room 103 of Figure 7.6. 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 LANs.

PRACTICE PROBLEMS

1. Explain why message collisions would or would not occur on local area networks that used the ring topology of Figure 7.4(b) or the star topology of Figure 7.4(c).
2. 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**.

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



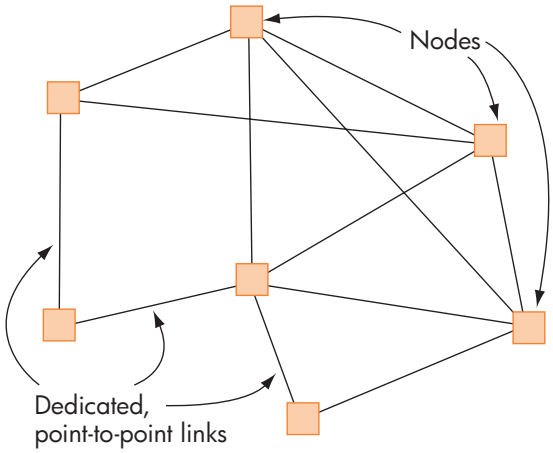
Explain exactly how node A on LAN1 sends a message to node B on LAN3.

7.2.3 *Wide Area Networks*

A **wide area network (WAN)** connects devices that are not in close proximity but rather are across town, across the country, or across the ocean. Because WANs cross public property, users must purchase telecommunications services, like those described in Section 7.2.1, from an external provider. Typically, these are **dedicated point-to-point** lines that directly connect two machines, and not the shared channels found on a LAN such as Ethernet. The typical structure of a WAN is shown in Figure 7.7.

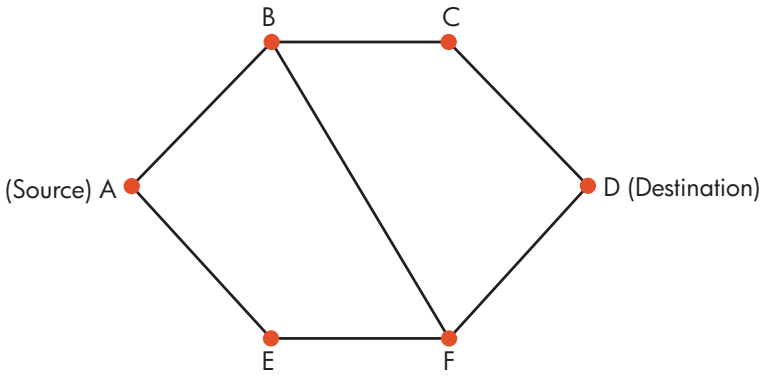
Most WANs use a **store-and-forward**, **packet-switched** technology to deliver messages. Unlike a LAN, in which a message is broadcast on a shared channel and is received by all nodes, a WAN message must "hop" from one node to another to make its way from source to destination. The unit of transmission in a WAN is a **packet**—an information block with a fixed maximum size that is transmitted through the network as a single unit.

FIGURE 7.7
Typical Structure of a Wide Area Network



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 following 6-node WAN:



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 is large, it may be broken up into multiple packets, and each one may take a different route.

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 the previous diagram 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 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 two classes of networks, LANs and WANs, but all real-world networks, including the Internet, are a complex mix of both 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 building LAN, an administration building LAN, and so forth. These individual LANs might then be interconnected into a wide area "company network" that allows users to send e-mail 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 7.5(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 (as discussed in the previous section), or from an Ethernet LAN to a packet-switched, store-and-forward WAN. We can see this type of interconnection structure in Figure 7.8(a).

The configuration in Figure 7.8(a) 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 Web pages 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 world through an **Internet Service Provider**, or **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 computer to the Internet. This access occurs through a WAN owned by the ISP, as shown in Figure 7.8(b). 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 hundreds of millions of bits per second.

The scope of networking worldwide is so vast, 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 7.8(c).

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

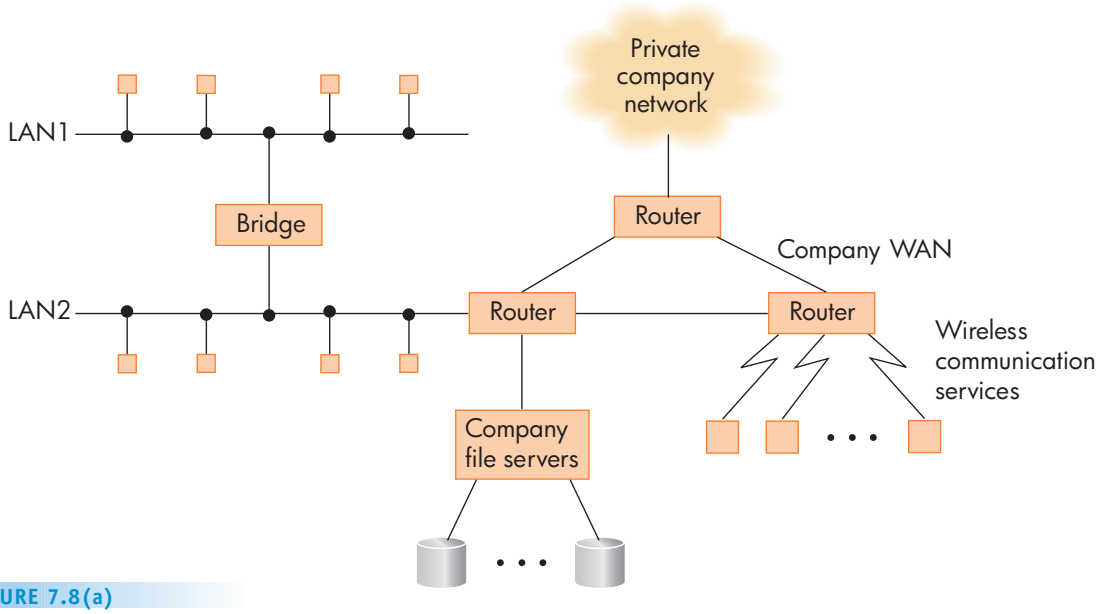
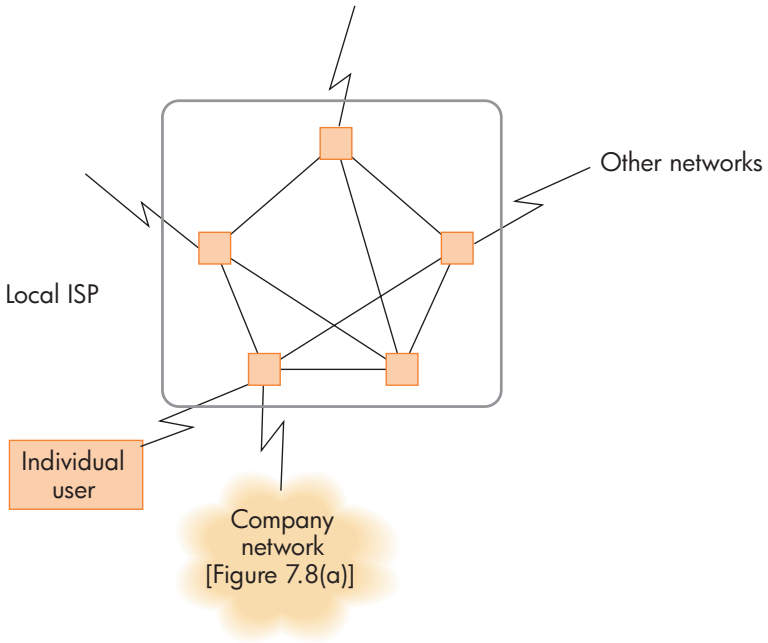


FIGURE 7.8(a)

Structure of a Typical Company Network

FIGURE 7.8(b)

Structure of a Network Using an ISP



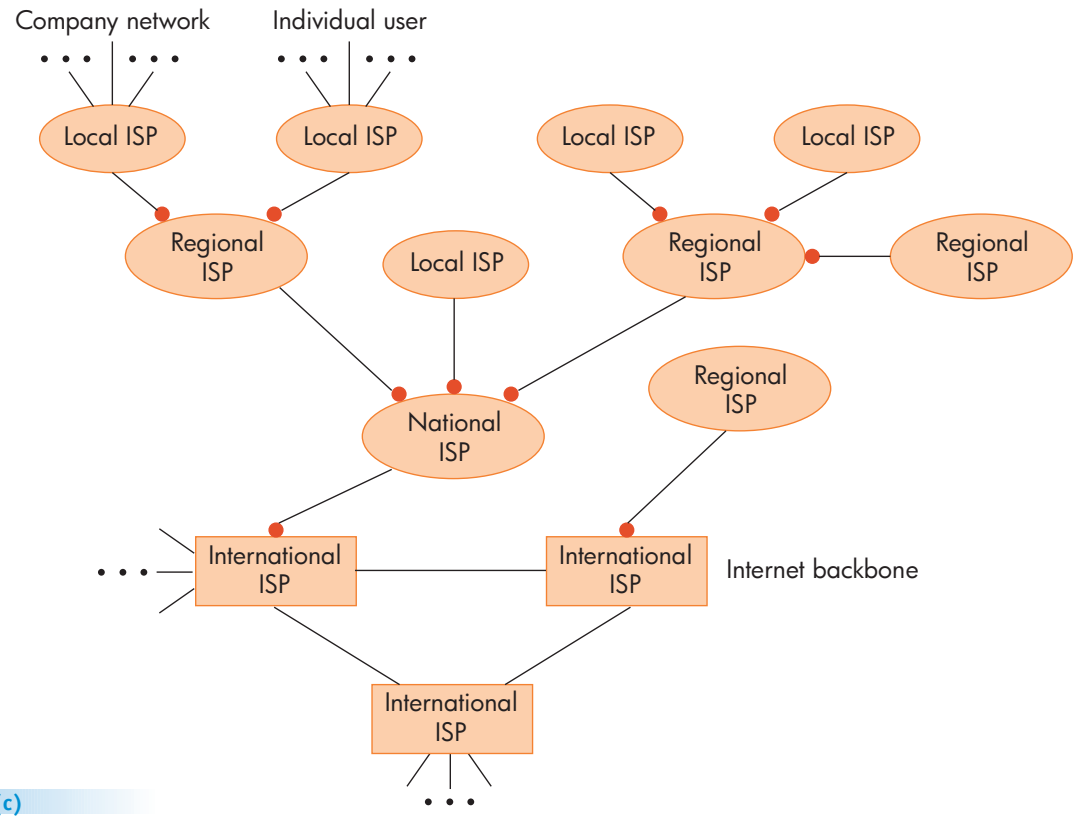


FIGURE 7.8(c)

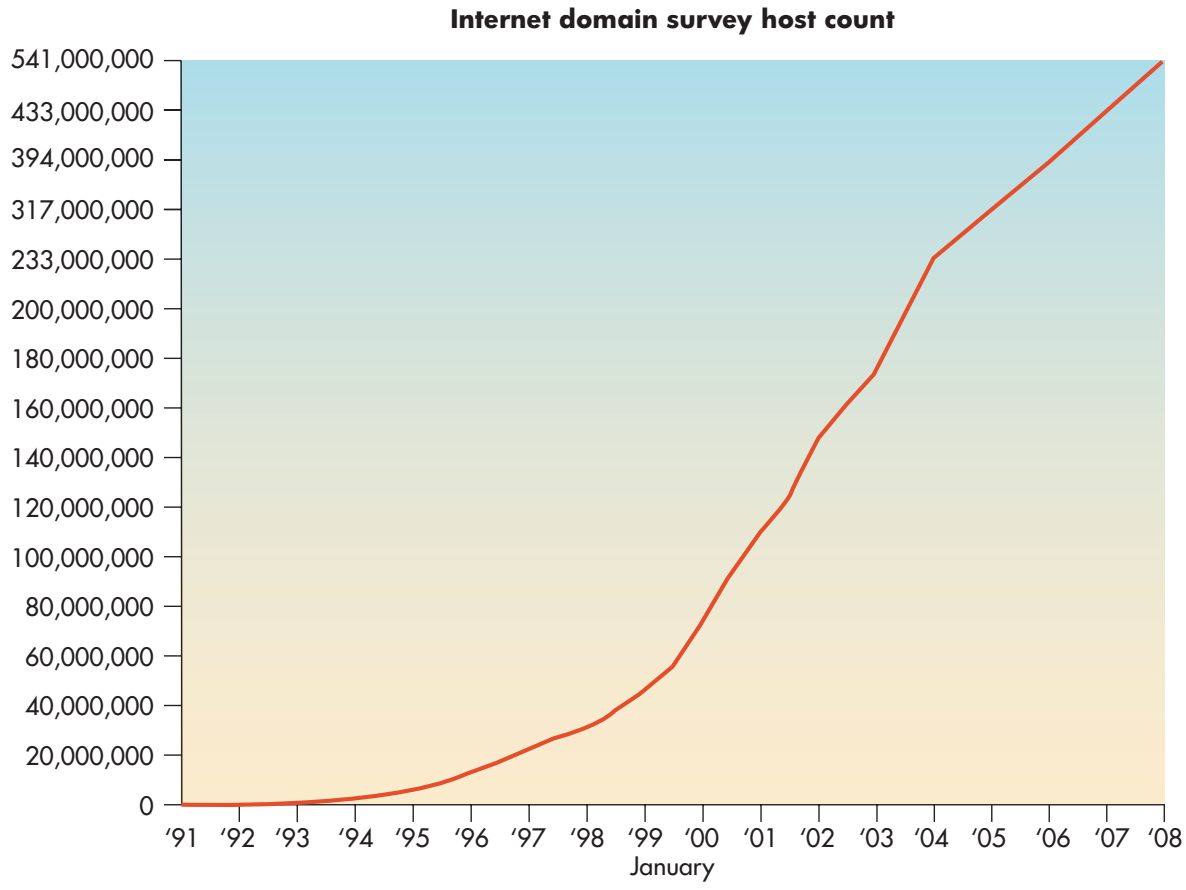
Hierarchy of Internet Service Providers

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, the telephone line from your home or office connects

to a local phone 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 7.8(c) 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, WANs, bridges, routers, and multiple levels of ISPs.

As of early 2008, there were about 541 million nodes (hosts) and hundreds of thousands of separate networks located in more than 225 countries. A graph of the number of host computers on the Internet over the last 17 years is shown in Figure 7.9. (This figure is really an undercount, because there are numerous computers located behind protective firewalls that do not respond to any external attempts to be counted.)

How does something as massive as the Internet actually work? How is it possible to get 541 million machines around the world to function efficiently as a single system? We answer that important question in the next section.



Source: Internet Software Consortium (http://www.isc.org/)

FIGURE 7.9
Internet Domain Host Survey Count Graph

7.3 Communication Protocols

When you talk on the telephone, there is a set of procedures that you follow. 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"—these conventions are what allows orderly exchanges to take place. Imagine what would happen if someone were unaware of them. For example, such a person might answer the phone but not say anything. Hearing silence, the person on the other end 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 hundreds of millions of machines made by dozens of manufacturers and located in hundreds of countries, they can all exchange messages correctly and efficiently for one simple reason: They all agree 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 more than 100 worldwide organizations (e.g., foundations, governmental agencies, educational institutions, companies) in 180 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 governmental 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 activities, check out its home page at www.isoc.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 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 7.10. This hierarchy is also referred to as **TCP/IP**, after the names of two of its most important protocols.

In the following sections we briefly describe the responsibilities of each of the five layers in the hierarchy shown in Figure 7.10.

FIGURE 7.10
The Five-Layer TCP/IP Internet Protocol Hierarchy

Layer	Name	Examples	
5	Application	HTTP, SMTP, FTP	
4	Transport	TCP, UDP	
3	Network	IP	
2b	Logical Link Control	PPP, Ethernet	Data Link Layer
2a	Medium Access Control	Ethernet	
1	Physical	Modem, DSL, Cable Modem, Wi-Fi, 3G	

7.3.1 *Physical Layer*

The **physical layer protocols** govern the exchange of binary digits 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
- What voltage levels are used to represent a binary 0 and a binary 1
- The shape of the connector between the computer and the 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 7.11.

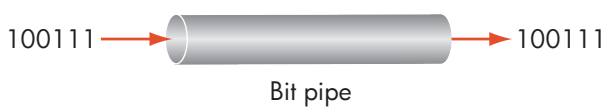
Once you select a physical layer protocol by purchasing a modem, getting a digital subscriber line, or using a mobile phone with wireless data capabilities, you can transmit binary signals across a physical channel. From this point on in the protocol stack, you no longer need be concerned about such 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 presents you with a 0 or a 1.

7.3.2 *Data Link Layer*

The physical layer protocols create a bit pipe between two machines connected by a communications link. However, this link is not an error-free channel, and due to interference or weather or any number of other factors, it can introduce errors into the transmitted bit stream. The bits that come out may 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, and not 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 the **framing** problem. 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.

FIGURE 7.11
The Concept of a Bit Pipe



In Section 7.2.1 we described how local area networks communicate by having multiple machines connected to a single shared communication line (Figures 7.5 and 7.6). However, while 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 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 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 wishing to send monitors the status of the line and, as soon as it becomes idle, it transmits. This situation is diagrammed in Figure 7.12(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.

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 7.12(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. This situation is diagrammed in Figure 7.12(c).

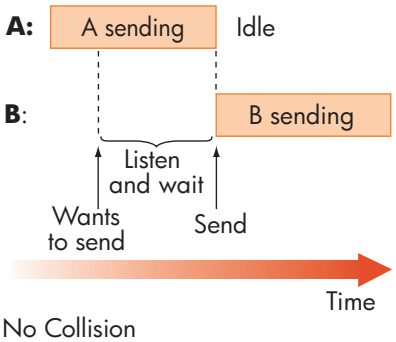
One reason Ethernet is so popular is that control is *distributed*. Responsibility for network operation is shared by all nodes in the network rather than centralized in a single master controller. Each node makes its own decisions about when to listen, when to send, and when to wait. That means that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

If our network uses point-to-point links like those in Figure 7.7, rather than shared lines, we 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 we are using a shared channel or a point-to-point link, we now have a sender and a receiver, who want to exchange a single message, directly connected by a channel. It is the job of the layer 2b **logical link control protocols** to ensure that the message traveling across this channel from source to destination arrives correctly.

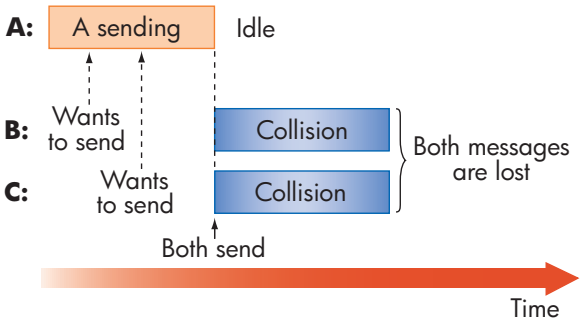
How is it possible to turn an inherently error-prone bit pipe like the one in Figure 7.11 into an error-free channel? In fact, we cannot entirely eliminate

FIGURE 7.12

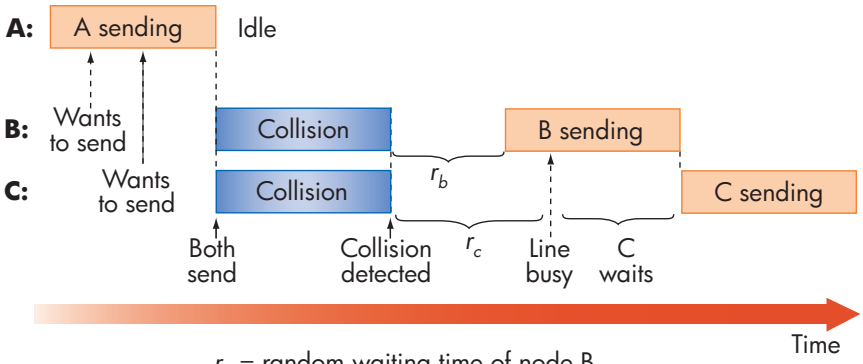
The Medium Access Control Protocols in Ethernet



(a) No Collision



(b) Collision



(c) Collision and Retransmission

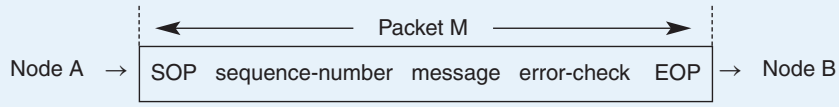
errors, but we can detect that an error has occurred and retransmit a new and unblemished copy of the original message. The **ARQ algorithm**, for *a*utomatic *r*epeat *req*uest, is the basis for all data link control protocols in current use.

Remember that at this point, nodes A and B are directly connected by a physical link. When A wishes to send a message to B, it first adds some additional information to form what is called 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 like Figure 7.13. 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

FIGURE 7.13

A Message Packet Sent by the Data Link Protocols



packet, it examines the error-check field to determine if the packet was transmitted correctly.

What makes the ARQ algorithm work is that 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 now 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, 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
		No response. Wait for a while
M(2) →		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 correct and in the proper sequence.

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*
1.	M(3)	→	
		←	ACK(3)
	M(3)	→	
			?
2.	M(3)	→	
		←	ACK(3)
	M(4)	→	
			?
3.	M(3)	→	
		←	ACK(3)
	M(5)	→	
			?

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 7.7, 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. As part of this delivery task, every node must agree to use the same node addressing scheme so that everyone is able to identify that ultimate destination. Thus, the two critical responsibilities of the network layer are

- Creating a universal addressing scheme for all network nodes
- 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 ***I*nternet *P*rotocol**.

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

macalester.edu

hawaii.edu

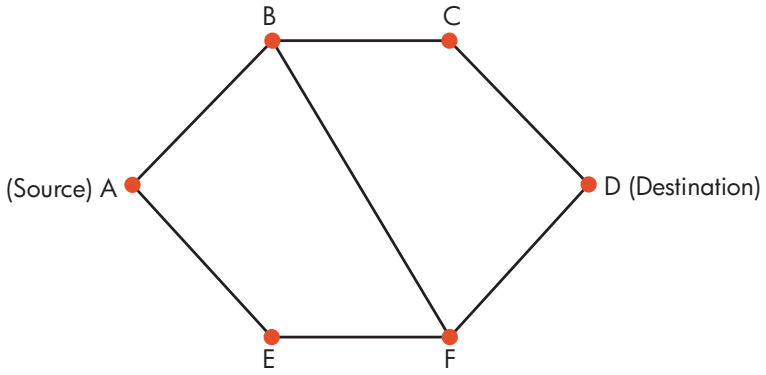
However, these **host names** are not the actual names that nodes use to identify each other in IP. Instead, nodes identify each other using a 32-bit **IP address**, often written as four 8-bit numeric quantities in the range 0–255, each grouping separated by a dot.[1] For example, the machine referred 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 that is 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 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 540 million or so host computers on the Internet. When you use a symbolic host name, such as mySchool.edu, 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 IP address for this symbolic name. If so, it returns the corresponding 32-bit value. If not, the local name server forwards it on to a remote name server (and possibly another, and another, . . .) until it locates the name server that knows the correct IP address.

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



Assume A wishes to send a message to D. First, node A uses the DNS to obtain the 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**.

---

1 The people who assign IP addresses are actually starting to run out of numbers. The new standard for IP, called IP version 6, increases the size of the address field from 32 to 128 bits. This will provide enough IP addresses for every atom in the universe, and then some! They are determined not to run out of addresses this time!

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. 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 example above, 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, but for networks like the Internet, with hundreds of millions 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.

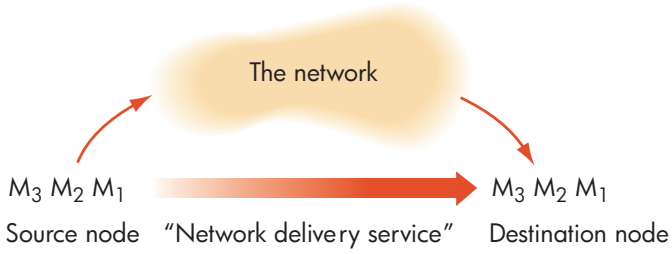
However, in Section 3.5 we showed that, as the number of network nodes increases, the solution time for these "brute force" algorithms grows exponentially. Therefore, this method is infeasible for any but the tiniest networks. Fortunately, there are much better algorithms that can solve this problem in $\Theta(N^2)$ 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 $N = 10^8$ or 10^9, an $\Theta(N^2)$ algorithm might require on the order of 10^{16} or 10^{18} calculations to determine the best route from any node to another—still an enormous amount of work.

There are additional problems that make routing difficult. One complication is *topological change*. The Internet is highly dynamic, with new links and new nodes added on an almost 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 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.

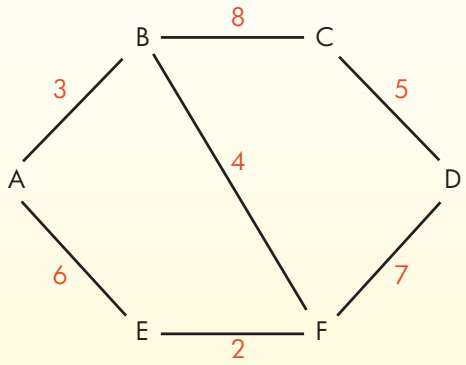
The network layer has many other responsibilities not mentioned here, including network management, broadcasting, and locating mobile nodes that move around the network. The network layer is truly a complex piece of software.

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 "network delivery service" in which messages are delivered between any two nodes in the network, regardless of their location:



PRACTICE PROBLEMS

Given the following 6-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 there is more waiting), answer the following questions:



1. 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.
2. Do you think the algorithm you used in 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?
3. 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?

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

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.

MyTown, Minnesota

The same situation exists on the Internet. Every host computer has an IP address that uniquely identifies it. However, there may be many 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**. While 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 7.14. 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, with port numbers 12 and 567 respectively, while Host B is executing two programs named Y and Z, with port numbers 44 and 709.

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 7.14, 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

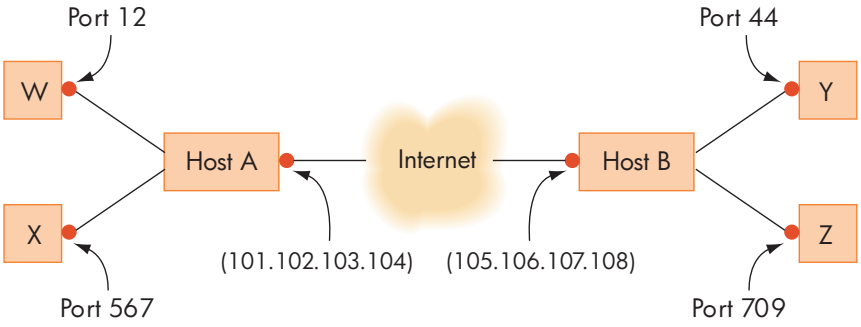


FIGURE 7.14
Relationship between IP Addresses and Port Numbers

network? The answer is that all important applications on the Internet use **well-known port numbers**. Just as it is widely known in the U.S. 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 Web pages (and which we discuss in the following section), always uses port 80. If you wish to get a Web page from another machine, you simply need to talk to the program that is listening for messages on port 80.

Figure 7.16 lists the port numbers of some common Internet applications. A list of all well-known port assignments is contained in the report titled *Assigned Numbers on the Internet* (RFC 1700) available over the Internet.[2] The only time you need to get a new port number is when you are developing a new application.

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 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, they do not guarantee that absolutely every letter you send will arrive, and they do 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**, an acronym for *T*ransport *C*ontrol *P*rotocol. (There is another transport protocol called **UDP** for *U*ser *D*atagram *P*rotocol. We will not be discussing it here.)

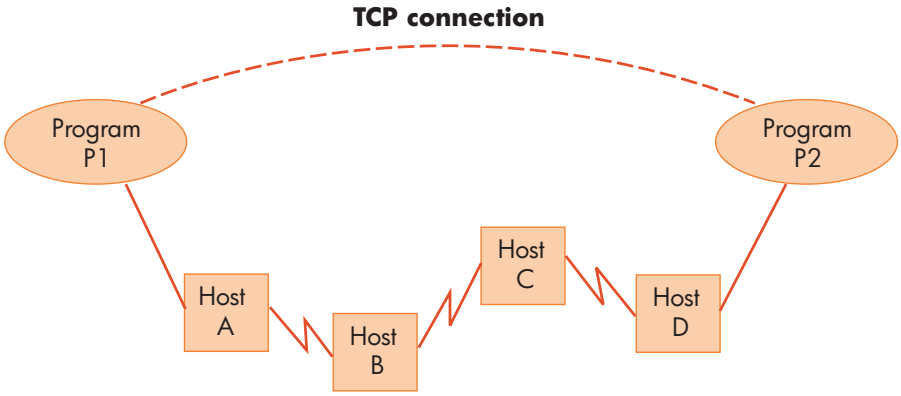
TCP requires that the two programs at the source and destination node initially establish a **connection**. That is, they must first inform each other of the impending message exchange, and they must describe the "quality of service" they wish to receive. This connection does not exist in a hardware 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 7.15.

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 go 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.

---

2 RFCs, as we mention in the Challenge Work section, are technical documents that describe virtually all aspects of Internet behavior. They are all available online at *http://www.faqs.org/rfcs/*

FIGURE 7.15
Logical View of a TCP Connection



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 four protocol layers in place, we 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 do it both correctly and efficiently. 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, What types of applications do we want to give to our network users and exactly how do we implement them? We answer that question as we look at the very top layer of our protocol stack—the application layer.

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 exciting new applications (often called **"killer apps"**) has fueled the rapid growth of networking and the Internet—e-mail in the 1970s, chat rooms in the 1980s, the Web and e-commerce in the 1990s, and Internet-based applications and collaborative computing in the twenty-first century. Figure 7.16 lists a few of the important application protocols on the Internet.

It is not possible in this one section to discuss all the protocols listed in Figure 7.16. Instead, we will use the HTTP protocol, which is used by the World

FIGURE 7.16

Some Popular Application Protocols on the Internet

Acronym	Name	Application	Well-known Port
HTTP	Hypertext Transfer Protocol	Accessing Web pages	80
SMTP	Simple Mail Transfer Protocol	Sending electronic mail	25
POP3	Post Office Protocol	Receiving electronic mail	110
IMAP	Internet Mail Access Protocol	Receiving electronic mail	143
FTP	File Transfer Protocol	Accessing remote files	21
TELNET	Terminal Emulation Protocol	Remote terminal access	23
DNS	Domain Name System	Translating symbolic host names to 32-bit IP addresses	42

Wide Web to access and deliver Web pages, to serve as a general model for how application layer services are typically built on top of the TCP/IP protocol stack.

A single Web page 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 page. The most common format is hypertext, and we access it using the **hypertext transfer protocol** called HTTP. (The Web is designed to accept and transfer other types of information as well. Thus, we could use the protocol identifier "news" to obtain information from bulletin boards and news groups, or "mailto," which allows us to send and receive e-mail documents via the Web.) 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/history.html

This identifies a hypertext ("http") document stored in a file called /about/history.html 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 simply *www.macalester.edu/about/history.html*.)

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

Once we establish this connection, we use the HTTP application protocol to access the desired Web page. An HTTP **request message** is sent on the TCP connection from the client to the server, specifying the name of a Web page. 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 the requested page.[3]

Let's illustrate how these pieces work together using a simple example. Imagine that you are using a Web browser and have just clicked on the following URL:

http://www.macalester.edu/about/history.html

The following sequence of events 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 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, the browser scans the URL to identify the page you want to access. In this case it is /about/history.html. The browser constructs an http GET message, which requests the contents of that Web page. This GET message looks something like the following:

   GET /about/history.html HTTP /1.1

   Host: www.macalester.edu

   Accept-language: English

   This message says that we want a copy of the English language page /about/history.html located at *www.macalester.edu*, and it should be accessed using the HTTP protocol, version 1.1. (An actual GET message is a bit more complex and includes a number of additional fields not shown here.)
4. The http GET message in step 3 is transmitted across the Internet from the client's Web browser program at the source node to the Web server 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, it is delivered to the Web server program (which is listening on port 80). The Web server locates the file named in the GET message and creates a response message containing a copy of the contents of that file. This response message looks something like the following:

   HTTP/1.1 200 OK

   Connection: close

   Date: Thursday, 26 Mar 2009

   Content Length: 53908

   Content Type: text/html

   . . . (the contents of the Web page go here) . . .

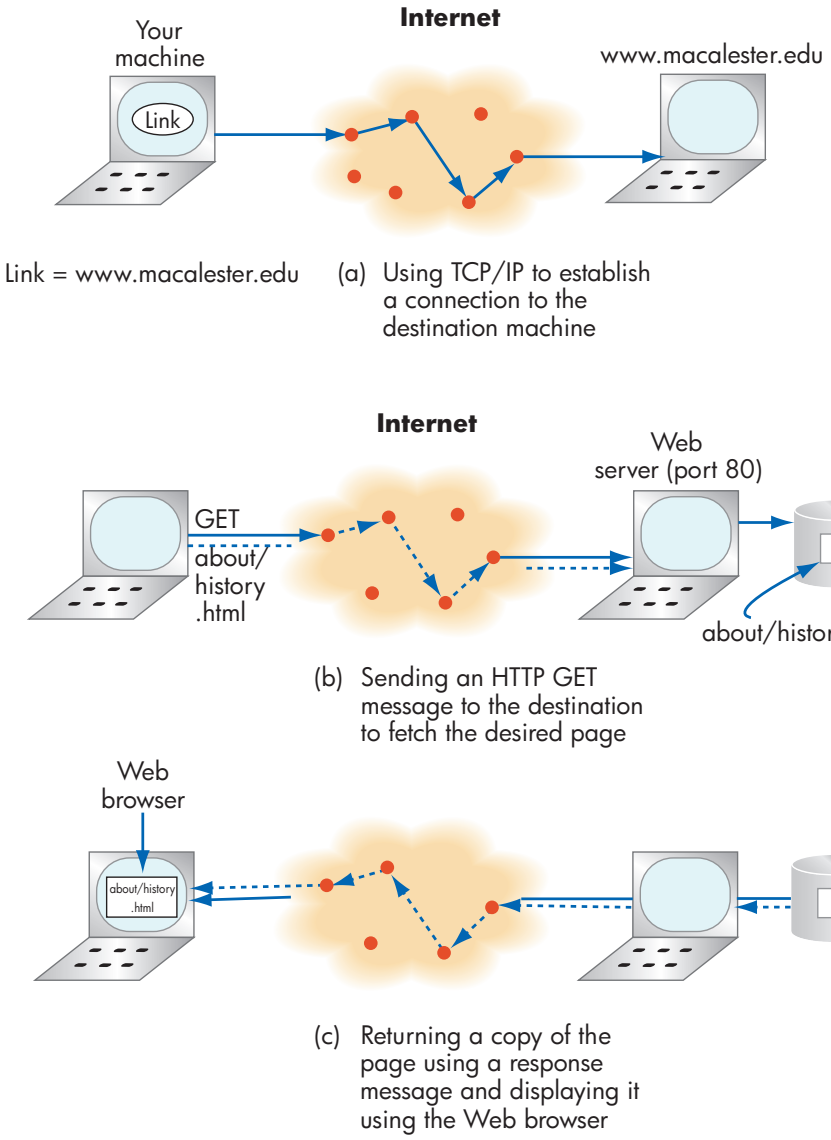
3 We have probably all had the experience of seeing return code 404: Page Not Found, which means the requested document does not exist on this server.

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 Web page has been sent, the TCP connection between the browser and the server will be closed. Finally, there is a copy of the entire Web page. (Again, some fields in the response message have been omitted for clarity.)

6. This 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 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 we click on a new URL. This sequence of events is diagrammed in Figure 7.17.

FIGURE 7.17
Behavior of the HTTP Application-Level Protocol



LABORATORY EXPERIENCE 11

We have just completed a rather long and complex discussion of how a computer network functions. The chapter presented a good deal of technical material that for some can be fairly difficult to grasp. To help clarify these ideas, this laboratory experience illustrates network behavior using a software package called a network simulator. This simulator allows you to observe and control many of the technical concepts introduced in this section, concepts such as packets, messages, error detection, error correction, and routing. Shown here is an example of the type of information you will be working with in this laboratory experience.

By simulating the behavior of a wide area network, many of the technical concepts introduced in the preceding pages should become more clear and understandable.

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 examine the services they offer and their impact on society.

Electronic mail (e-mail) has been the single most popular application of networks for the last 30 years. When the Internet was first developed, its designers thought that it would be an ideal way to access advanced, high-performance hardware and software. Instead, they found that it was a wonderfully effective way to communicate, and e-mail rapidly became the dominant application.

E-mail is *convenient*. You can send a message whenever you want, and it waits for the recipient to log on and read it at his or her convenience. E-mail is *fast*. A message from the United States typically arrives anywhere in the world in less than a minute, 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). E-mail supports *multimedia*. The contents of your electronic messages are not limited to characters but can also include a wide range of *attachments*, including photographs, text, graphics, and sound. Finally, e-mail is a *broadcast medium*. A computer can send a letter to a thousand recipients as easily as it can send it to one (which may, in fact, be a detriment rather than an advantage as we see in the box titled "Spam").

An interesting application related to e-mail is **bulletin boards**. A bulletin board is a shared public file where anyone can post messages and everyone is free to read the postings of others. It is an electronic version of the bulletin boards commonly seen in grocery stores, cafes, and public libraries. Most bulletin boards are associated with a particular topic or special area of interest. These specialized bulletin boards, called **news groups**, are a wonderful way to create a community of individuals who share a common interest and want to exchange ideas and opinions. Some news groups support **chat rooms**—the real-time exchange of messages. Rather than posting a message that is read at a later time, what the

Spam

Spam is electronic "junk mail"—unsolicited e-mail sent to thousands, even millions, of network users without their permission or knowledge. It is not certain where the term *spam* came from, but the best guess is from the famous Monty Python comedy routine in which the word *spam* is repeated over and over, making it a synonym for the seemingly endless repetition of silly or worthless words.

Junk mail in the form of advertising flyers and political brochures has been a staple of surface mail for many years. But there is a natural cap on its volume—the sender has to pay the post office to deliver it. So, if a mailing is not likely to produce a profitable return or have a worthwhile purpose, it will not be sent.

However, e-mail does not have that built-in cap, and it is beginning to clutter our electronic mail boxes and consume large amounts of bandwidth. Because the Internet is a public facility, there is little that can be done to regulate spam. A number of companies have developed **mail filters** that attempt to determine which e-mails are useful and which are spam. Unfortunately, as soon as a filter is developed and released, the "spammers" quickly figure out a way to beat it and get their e-mail through to your machine. Probably the best filter developed so far (and for the near future) is the DELETE button on your keyboard!

sender types appears immediately on the screen of one or more individuals, allowing for the direct exchange of ideas. Another popular form of real-time message exchange is **Instant Messaging (IM),** the rapid exchange of messages, often using wireless technology. This service is provided by many of the large ISPs, such as MSN, Yahoo, and AOL. All these ways of keeping in touch with other people (e-mail, chat, IM) have led to the development of what are called **social networks**—systems that create communities of users who share common interests and activities and which provide multiple methods of online interaction. See the box on Social Networking on page 328.

Another important network service is **resource sharing**, the ability to share *physical resources*, such as a printer or storage device, as well as *logical resources*, such as software and information.

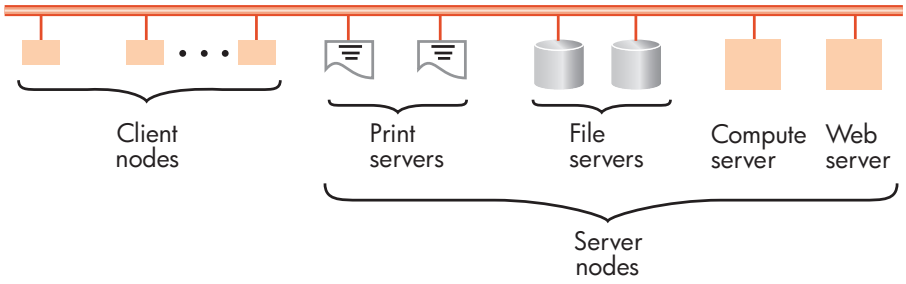
The prices of computers and peripherals have been dropping for many years, so it is tempting to think that everyone should buy their own I/O or storage devices. However, that is not a cost-effective way to configure computer systems. For example, a high-quality color printer may be used rather infrequently. Buying everyone in the office his or her own printer would leave most of them idle for long periods of time. It is 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 makes sense to keep a single copy on a shared network disk, called a **file server**. A network file server can also be a cost-effective way to provide shared backup services to multiple sites.

The style of computing wherein some nodes provide services while the remaining nodes are users (or clients) of those services is called, naturally enough, **client-server computing**. We have seen two examples—print servers and file servers—but there are many others, such as mail servers, name 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 7.18.

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

FIGURE 7.18

The Client-Server Model of Computing



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**. Web pages can be exchanged between remote systems. Files can be transmitted anywhere in the world using FTP, which is mentioned in Figure 7.16, and online databases can be accessed by authorized users regardless of location.

Many network sites now provide a service called an **information utility**, also known as a **data warehouse**. These nodes contain massive amounts of information that can be electronically searched for specific facts or documents. Frequently such sites contain highly specialized information, such as geopolitical data, current stock prices, real estate records, or information on case law and legal precedents. Nowadays it is more common for students, scientists, businesspeople, and politicians to search for information at their monitor than in the stacks of a library.

Another important network 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 in progress online. A rapidly growing network application is **collaborative software**, also known as **groupware**—software that facilitates the efforts of individuals connected by a network and working on a single shared project.

Electronic commerce (or just **e-commerce**) 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 e-commerce 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 check-out 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 World Wide 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 5–10 years, traffic on the Internet has changed from primarily academic and professional to heavily commercial. For example, as of early 2008, there were about 95,000,000 host computers in the .com (U.S. commercial) domain, while fewer than 11,000,000 were in the .edu domain (U.S. educational institutions).

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

7.5 A Brief 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 World Wide Web.[4]

In the words of its designers, "The Internet has revolutionized the computer and communications world like nothing before. It is at once a worldwide broadcasting capability, a mechanism for information dissemination, and a medium for collaboration and interaction between individuals and their computers without regard for geographic location."

7.5.1 *The Internet*

The Internet is not a recent development but an idea that has been around for more than 40 years. The concept first took shape during the early and mid-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 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. (Traditional dial-up telephones were considered too vulnerable because the failure of the central phone switch would completely cut all voice communications. As we described earlier, a WAN can automatically route around a failed line or node in order to maintain communications.)

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 packet-switched 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. Later that same year two more nodes were

4 Leitner, B., Cerf, V., Kahn, R., Kleinrock L., Lynch, D., Postel, J., Roberts, L., and Wolff, S., "A Brief History of the Internet," Version 3.32, *www.isoc.org/internet/history/brief.shtml*, December 10, 2003.

added (the University of California–Santa Barbara and the University of Utah), and by the end of 1969, the budding 4-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" (critically important application) 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-to-machine traffic that dominated usage in the first few years. (It is interesting to note that the total e-mail volume in the United States in 2007 was about 10 trillion messages!)

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 specific 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 the IBM Corp. and DECNet from the Digital Equipment Corporation. The 1970s were a time of rapid expansion of networks in both the academic and commercial communities.

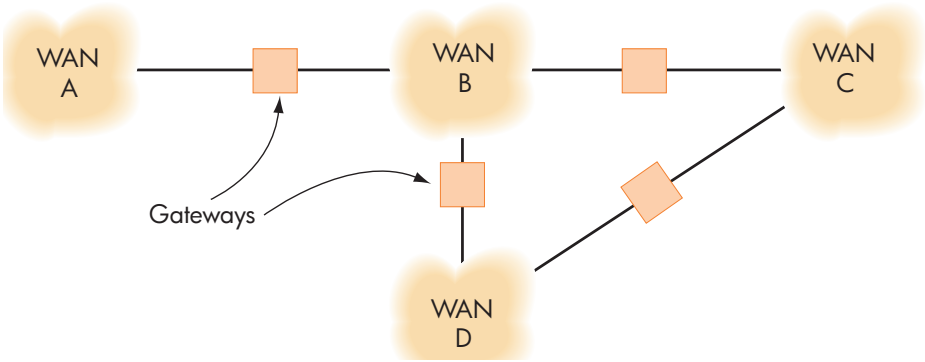
Farsighted researchers at ARPA, in particular Robert Kahn, realized that this rapid and unplanned proliferation of independent networks would lead to incompatibilities and prevent users on different networks from communicating with each other, a situation that brings to mind the problems that national railway systems have sharing rail cars because of their use of different gauge track. Kahn knew that to obtain maximum benefits from this new technology, all networks would 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 7.19 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 7.19

A Network of Networks



To allow the four WANs of Figure 7.19 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 Dr. 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 on the PCs and desktop machines that were just 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 the modern computing environment—desktop PCs connected by LANs.

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. At the same time, exciting new applications appeared that were designed to meet the growing needs of the networking community. (Many of these application protocols were introduced in Section 7.3.5.) For example, **Telnet** is a software package that allows users to log on remotely to another computer and use it as though it were their own local machine. **FTP (file transfer protocol)** provides a way to move files around the network quickly and easily. Along with e-mail (still wildly popular), these and other new applications added more fuel to the superheated growth of computer networks.

With TCP/IP becoming a de facto network 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. 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 [or to have?] a significant impact on the general population for one very important reason: In order to use the ARPANET, you needed a research grant from the U.S. Department of Defense (DOD). By the early 1980s, many people were using the Internet, but they were almost exclusively physicists, engineers, and computer scientists at a select set of secure military and 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. 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 medical centers, schools, research centers, and government agencies. As these national networks were created, they were also 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 7.20.

Some time in the late 1980s, the term ARPANET ceased to be used because, as Figure 7.20 shows, the ARPANET was now only one of many networks belonging to a much larger collection. (By 1990, it had grown to 300,000 computers on 3,000 separate networks.) People began referring to this entire collection of interconnected networks as "the Internet," though this name was not officially accepted by the U.S. government until October 24, 1995.

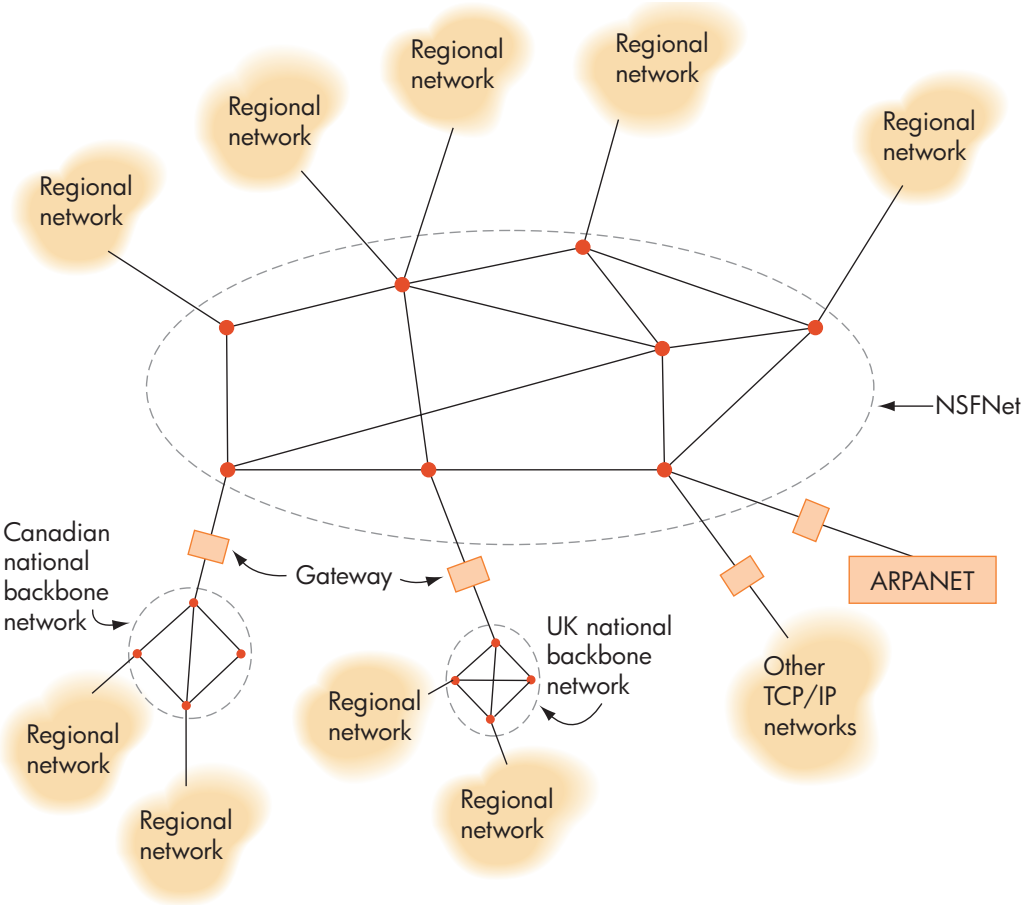


FIGURE 7.20
State of Networking in the Late 1980s

Once people 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** that offered the Internet access once provided by the ARPANET and NSFNet.

By early 2008, the Internet had grown to 541,000,000 computers located in just about every country in the world. The extraordinary growth of the Internet continues to this very day. Figure 7.9 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 40 years, into a global communications infrastructure moving trillions of bits of data among hundreds of millions 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 PCs and local area networks), and changes in scale (from hundreds of nodes to hundreds of millions).

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

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

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), Bouvet Island (.bv), Mayotte (.yt), Kiribati (.ki), Svalbard and Jan Mayen Islands (.sj), and even the continent of Antarctica (.aq), which includes more than 100 computers in its domain. The smallest non-empty DNS domain is .wf—the Wallis and Futuna Islands, a tiny French territory in the South Pacific between Hawaii and New Zealand. As of early 2008 it contained exactly one host computer!

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 7.21. 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 ***U*niform *R*esource *L*ocator**, and it is the worldwide identification of a Web page located on a specific host computer on the Internet.

Berners-Lee named his new information system the **World Wide Web**, and it was completed and made available to all researchers at CERN in May 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.

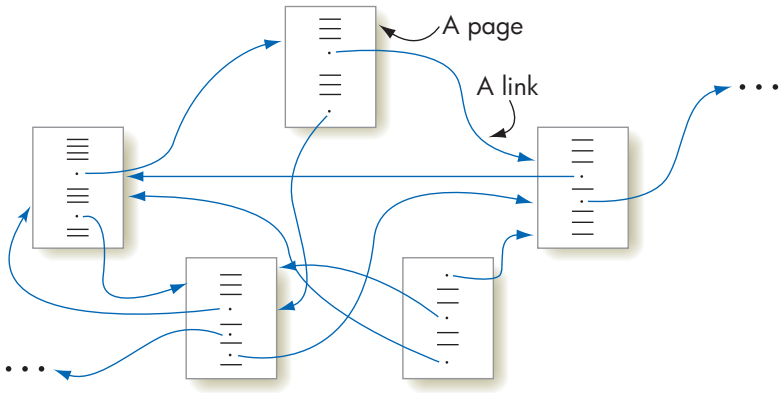
A powerful graphical Web browser, called Mosaic, was developed in late 1993 and made available to the general public so that they could begin to use this new service. With the appearance of Mosaic, the World Wide Web began to "take off." It was a network application that offered users exactly what they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

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 108 million distinct Web sites and about 8–12 billion pages by early 2008. It is by far the fastest growing component of the Internet. The Web's colorful graphics and simple point-and-click method of accessing

FIGURE 7.21

Hypertext Documents



information has made it the Internet killer app of the twenty-first century. It has become the vehicle for bringing the capabilities of networking to everyone—from toddlers to senior citizens and kindergarten students to PhDs. For many people, the World Wide Web *is* the Internet.

7.6 Conclusion

Computer networking has changed enormously in the 40 or so 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 hundreds of millions, 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.

Social Networking

Computer networks were originally created to provide scientists and engineers with easy access to important software packages and data files stored on remote computers. However, the first Internet "killer app" was rather unexpected and something quite different—e-mail—and while many messages did contain technical material, 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 e-mail in the early 1970s, there were many other attempts to foster online communities. The first **bulletin board system (BBS)** appeared in 1978. It allowed users to dial a central site using a time-sharing terminal, read and post notices, chat, play online games, and exchange messages. A similar system, called **Usenet,** was developed in 1980. It was similar to a BBS with the added feature of supporting "newsgroups"—subgroups of users who indicate an interest in a particular topic, such as space flight, Chinese cooking, or Minnesota Vikings football. Usenet subscribers could post notices to and chat with members of just one specific newsgroup. BBS systems were very popular from the late 1970s until the mid-1990s when they began to be replaced by Web-based applications. One of the earliest examples of using the Web to support interpersonal communication and collaboration was the wiki. A **wiki** is a set of Web pages that everyone is free to access, add to, or modify. It is essentially a collaborative shared document built and maintained by a community of online users. The most well known and widely used wiki is the online encyclopedia Wikipedia, which currently has over 2.5 million English-language articles, not to mention millions of other articles in languages from Polish to Portuguese, from French to Finnish. (By comparison, the *Encyclopedia Britannica* has about 0.5 million articles.)

The use of Web-based social networking sites has grown to the point where they are some of the most well known and widely used applications on the Internet, and there is hardly a young person today who is not thoroughly familiar with them and, more likely, a registered member. **Facebook,** developed in 2004 by Mark Zuckerberg while a student at Harvard, has 36 million subscribers in the U.S. alone and receives 132 million visits per month. **MySpace,** which appeared one year earlier in 2003, has 73 million subscribers and 117 million visitors per month. **LinkedIn,** also started in 2003, has 24 million registered users.

Twitter, another popular social networking tool, allows users to keep up with those in their circle of friends with "tweets," short text-based posts that appear on the users' and friends' pages. By some estimates, Twitter is now the third largest social network, after Facebook and MySpace.

The most pressing issue facing the Internet today is not technology and new applications. Those issues have been and will continue to be addressed and solved by the computer science community. 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. Currently, the Internet is managed by the *Internet Society*, the nonprofit agency first introduced in Section 7.3. Now that it is a global phenomenon that affects millions 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 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 page 322) on the history of networking:

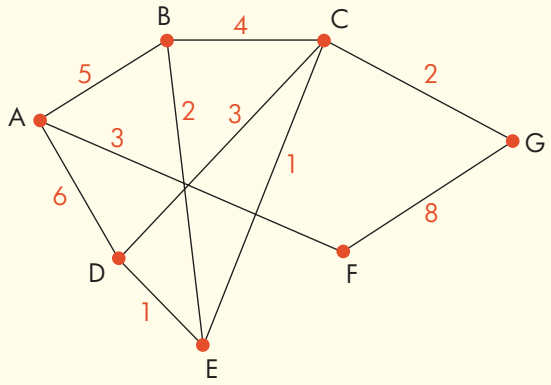
> If the Internet stumbles, it will not be because we lack for technology, vision, or motivation. It will be because we cannot set a direction and march collectively into the future.

EXERCISES

1. Show how a modem would encode the 5-bit binary sequence 11001 onto an analog carrier by
   a. Modifying its amplitude (the height of the carrier wave)
   b. Modifying its frequency (the number of waves per second)
2. A modem can also modify the *phase* of a carrier wave to encode binary data. Find out what the phase of a signal is and determine how it can be modified so that it can encode the same 5-bit signal 11001 used in Exercise 1.
3. Determine the total time it takes to transmit an uncompressed grayscale image (with 8 bits/pixel) from a screen with a resolution of 1,280 $\times$ 840 pixels using each of the following media:
   a. A 56 Kbps modem
   b. A 1.5 Mbps DSL line
   c. A 100 Mbps Ethernet link
4. a. Assume there are 1 million books in your campus library. Approximate (to the nearest order of magnitude) how many bytes of data there are if all these books were stored online and accessible across a computer network.
   b. How long does it take to transfer the entire collection of books if the data rate of the transmission medium is 10 Mbps, the speed of the original Ethernet? How long does it take if we have a line with a speed of 1 Gbps? (This value represents the time needed to download your entire campus library.)
5. Why is the address field needed in an Ethernet LAN protocol? Can you think of a useful situation where you might want either to omit the address field entirely or to use some "special" address value in the address field?
6. After reviewing the description of the Ethernet protocol in Section 7.3.2, how do you think this protocol behaves in a very heavily loaded network—that is, a network environment where there are lots of nodes attempting to send messages? Explain what behavior you expect to see and why.
7. The Ethernet is a distributed LAN protocol, which means that there is no centralized control node and that the failure of a single node can never bring down the entire network. However, can you think of any advantage to the creation of a centralized LAN in which one node is in charge of the entire network and makes all decisions about who can send a message and who must wait? Explain.
8. Agree or disagree with the following assertion and state why:

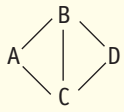
   In an Ethernet network, even though there are collisions, every message is guaranteed to be delivered in some maximum amount of time T.

9. a. Assume there is a wide-area network with N nodes, where $N \geq 2$. What is the *smallest* number of point-to-point communication links such that every node in the network is able to talk to every other node? (*Note:* A network in which some nodes are unable to exchange messages with other nodes because there is no path between them is called *disconnected*.)
   b. If you are worried about having a disconnected network, what type of interconnection structure should you use when configuring your network?
10. What happens to the store-and-forward protocol of Figure 7.8 if a packet M is repeatedly sent from node A to node B but never correctly arrives at B? (Perhaps the link from A to B is broken.) What modifications can we make to this protocol to handle this situation?
11. The ARQ algorithm described in Section 7.3.2 is quite inefficient because the sending node must stop sending until it receives an explicit ACK from the receiving node. Can you design a modification to the protocol that makes it more efficient, and not cause the sender to have to stop each time it sends a message? Describe your revised protocol in detail.
12. How do we *broadcast* a message using an ARQ algorithm? That is, how do we send the same message to 100 different nodes on a WAN?
13. Given the following diagram, where the numbers represent the time delays across a link:



   a. How many simple paths (those that do not repeat a node) are there from node A to G?
   b. What is the *shortest path* from node A to node G? What is the overall delay?
   c. If node E fails, does that change the shortest path? If so, what is the new shortest path?
14. What are some of the specific responsibilities performed by the device called a gateway (diagrammed in Figure 7.19) that is placed between two different types of networks to allow them to communicate?

15. In Section 7.3.4 we said that the transport layer turns the inherently unreliable network layer into an error-free delivery service. However, the network layer uses the services of the data link layer, which is guaranteed to correctly deliver messages on a point-to-point link. For example, assume we have the following 4-node network:



If the network layer is sending a message from A to D via B, it can be sure that a message sent by the data link layer from A to B will always correctly get to B, and a message sent from B to D will always correctly get to D. How then is it possible for the network layer to be unable to correctly deliver a message from A to D?

16. Look at the home page of the Internet Society (*www.isoc.org*) and read about one of the designers of the original ARPANET—Larry Roberts, Leonard Kleinrock, Vinton Cerf, Robert Kahn, John Postel, or others. Learn about the early days of networking and the contributions that these individuals made to the ultimate development of the Internet. The home page of the Internet Society has links to many other places that provide a wealth of fascinating information about networks in general and the Internet and the Web in particular.

CHALLENGE WORK

The TCP/IP protocols are the heart and soul of the Internet, and they describe the fundamental rules that govern all communications in the network. Read more about the TCP/IP protocols and write a report describing their basic characteristics and giving a simple overview of the way that they work.

One of the best places to go for this information is a set of documents called **RFCs (Request for Comments)**. These are a series of documents produced by the Internet Engineering Task Force (IETF) that describe virtually all aspects of the Internet's behavior, including its protocols. Some RFCs contain enormously detailed technical specifications of the workings of the Internet, while others are more informational or tutorial (even humorous) in nature. A good place to start is RFC 1180, "A TCP/IP Tutorial." A complete set of all the Internet RFCs is located at *http://www.faqs.org/rfcs*, and it can be searched using the searchable database located at that Web site.

FOR FURTHER READING

A number of texts provide good overviews of computer networking. For example, see:

Kurose, J. F., and Ross, K. *Computer Networking: A Top-Down Approach,* 4th ed. Reading, MA: Addison Wesley, 2007.

Stallings, W. *Data and Computer Communications*, 8th ed. Englewood Cliffs, NJ: Prentice-Hall, 2006.

Tanenbaum, A. S. *Computer Networks*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

In the following book, its original creator describes the creation of the World Wide Web:

Berners-Lee, T., and Fischetti, M., *Weaving the Web: The Original Design and the Ultimate Destiny of the Web.* New York: Harper Business, 2000.

This text provides excellent discussions of TCP/IP, the basis for Internet communications.

Comer, D. E. *Internetworking with TCP/IP*, 5th ed. Vol. 1, *Principles, Protocols, and Architectures*. Englewood Cliffs, NJ: Prentice-Hall, 2005.

The following is another book on various topics within the field of computer networks:

Izzo, P. *Gigabit Networking*. New York: Wiley, 2000.

Spam