

New and Augmented Materials

Although the contents of each chapter have been updated, some new materials have also been added to this edition:

- ❑ *Peer-to-Peer paradigm* has been added as a new chapter (Chapter 29).
- ❑ *Quality of service* (QoS) has been augmented and added as a new chapter (Chapter 30).
- ❑ Chapter 10 is augmented to include the *forward error correction*.
- ❑ WiMAX, as the wireless access network, has been added to Chapter 16.
- ❑ The coverage of the transport-layer protocol has been augmented (Chapter 23).
- ❑ Socket-interface programming in Java has been added to Chapter 25.
- ❑ Chapter 28, on multimedia, has been totally revised and augmented.
- ❑ Contents of unicast and multicast routing (Chapters 20 and 21) have gone through a major change and have been augmented.
- ❑ The next generation IP is augmented and now belongs to Chapter 22.

Changes in the End-Chapter Materials

The end-chapter materials have gone through a major change:

- ❑ The practice set is augmented; it has many new problems in some appropriate chapters.
- ❑ Lab assignments have been added to some chapters to allow students to see some data in motion.
- ❑ Some applets have been posted on the book website to allow students to see some problems and protocols in action.
- ❑ Some programming assignments allow the students to write some programs to solve problems.

Extra Materials

Some extra materials, which could not be fit in the contents and volume of the book, have been posted on the book website for further study.

New Organization

This edition is divided into seven parts, which reflects the structure of the Internet model.

Part One: Overview

The first part gives a general overview of data communications and networking. Chapter 1 covers introductory concepts needed for the rest of the book. Chapter 2 introduces the Internet model.

Part Two: Physical Layer

The second part is a discussion of the physical layer of the Internet model. It is made of six chapters. Chapters 3 to 6 discuss telecommunication aspects of the physical layer.

Chapter 7 introduces the transmission media, which, although not part of the physical layer, is controlled by it. Chapter 8 is devoted to switching, which can be used in several layers.

Part Three: Data-Link Layer

The third part is devoted to the discussion of the data-link layer of the Internet model. It is made of nine chapters. Chapter 9 introduces the data-link layer. Chapter 10 covers error detection and correction, which can also be used in some other layers. Chapters 11 and 12 discuss issues related to two sublayers in the data-link layer. Chapters 13 and 14 discuss wired networks. Chapters 15 and 16 discuss wireless networks. Chapter 17 shows how networks can be combined to create larger or virtual networks.

Part Four: Network Layer

The fourth part is devoted to the discussion of the network layer of the Internet model. Chapter 18 introduces this layer and discusses the network-layer addressing. Chapter 19 discusses the protocols in the current version. Chapters 20 and 21 are devoted to routing (unicast and multicast). Chapter 22 introduces the next generation protocol.

Part Five: Transport Layer

The fifth part is devoted to the discussion of the transport layer of the Internet model. Chapter 23 gives an overview of the transport layer and discusses the services and duties of this layer. Chapter 24 discusses the transport-layer protocols in the Internet: UDP, TCP, and SCTP.

Part Six: Application Layer

Chapter 25 introduces the application layer and discusses some network programming in both C and Java. Chapter 26 discusses most of the standard client-server programming in the Internet. Chapter 27 discusses network management. Chapter 28 is devoted to the multimedia, an issue which is very hot today. Finally, Chapter 29 is an introduction to the peer-to-peer paradigm, a trend which is on the rise in the today's Internet.

Part Seven: Topics Related to All Layers

The last part of the book discusses the issues that belong to some or all layers. Chapter 30 discusses the quality of service. Chapters 31 and 32 discuss security.

Appendices

The appendices (available online at <http://www.mhhe.com/forouzan>) are intended to provide a quick reference or review of materials needed to understand the concepts discussed in the book. There are eight appendices that can be used by the students for reference and study:

- ❑ Appendix A: Unicode
- ❑ Appendix B: Positional Numbering System
- ❑ Appendix C: HTML, CSS, XML, and XSL
- ❑ Appendix D: A Touch of Probability
- ❑ Appendix E: Mathematical Review

- ❑ Appendix F: 8B/6T Code
- ❑ Appendix G: Miscellaneous Information
- ❑ Appendix H: Telephone History

References

The book contains a list of references for further reading.

Glossary and Acronyms

The book contains an extensive glossary and a list of acronyms for finding the corresponding term quickly.

Pedagogy

Several pedagogical features of this text are designed to make it particularly easy for students to understand data communication and networking.

Visual Approach

The book presents highly technical subject matter without complex formulas by using a balance of text and figures. More than 830 figures accompanying the text provide a visual and intuitive opportunity for understanding the material. Figures are particularly important in explaining networking concepts. For many students, these concepts are more easily grasped visually than verbally.

Highlighted Points

I have repeated important concepts in boxes for quick reference and immediate attention.

Examples and Applications

Whenever appropriate, I have included examples that illustrate the concepts introduced in the text. Also, I have added some real-life applications throughout each chapter to motivate students.

End-of-Chapter Materials

Each chapter ends with a set of materials that includes the following:

Key Terms

The new terms used in each chapter are listed at the end of the chapter and their definitions are included in the glossary.

Recommended Reading

This section gives a brief list of references relative to the chapter. The references can be used to quickly find the corresponding literature in the reference section at the end of the book.

Summary

Each chapter ends with a summary of the material covered by that chapter. The summary glues the important materials together to be seen in one shot.

Practice Set

Each chapter includes a practice set designed to reinforce salient concepts and encourage students to apply them. It consists of three parts: quizzes, questions, and problems.

Quizzes

Quizzes, which are posted on the book website, provide quick concept checking. Students can take these quizzes to check their understanding of the materials. The feedback to the students' responses is given immediately.

Questions

This section contains simple questions about the concepts discussed in the book. Answers to the odd-numbered questions are posted on the book website to be checked by the student. There are more than 630 questions at the ends of chapters.

Problems

This section contains more difficult problems that need a deeper understanding of the materials discussed in the chapter. I strongly recommend that the student try to solve all of these problems. Answers to the odd-numbered problems are also posted on the book website to be checked by the student. There are more than 600 problems at the ends of chapters.

Simulation Experiments

Network concepts and the flow and contents of the packets can be better understood if they can be analyzed in action. Some chapters include a section to help students experiment with these. This section is divided into two parts: applets and lab assignments.

Applets

Java applets are interactive experiments that are created by the authors and posted on the website. Some of these applets are used to better understand the solutions to some problems; others are used to better understand the network concepts in action.

Lab Assignments

Some chapters include lab assignments that use Wireshark simulation software. The instructions for downloading and using Wireshark are given in Chapter 1. In some other chapters, there are a few lab assignments that can be used to practice sending and receiving packets and analyzing their contents.

Programming Assignments

Some chapters also include programming assignments. Writing a program about a process or procedure clarifies many subtleties and helps the student better understand the concept behind the process. Although the student can write and test programs in any computer language she or he is comfortable with, the solutions are given in Java language at the book website for the use of professors.

Audience

This book is written for both academic and professional audiences. The book can be used as a self-study guide for interested professionals. As a textbook, it can be used for a one-semester or one-quarter course. It is designed for the last year of undergraduate study or the first year of graduate study. Although some problems at the end of the chapters require some knowledge of probability, the study of the text needs only general mathematical knowledge taught in the first year of college.

Instruction Resources

The book contains complete instruction resources that can be downloaded from the book site <http://www.mhhe.com/forouzan>. They include:

Presentations

The site includes a set of colorful and animated PowerPoint presentations for teaching the course.

Solutions to Practice Set

Solutions to all questions and problems are provided on the book website for the use of professors who teach the course.

Solution to Programming Assignments

Solutions to programming assignments are also provided on the book website. The programs are mostly in Java language.

Student Resources

The book contains complete student resources that can be downloaded from the book website <http://www.mhhe.com/forouzan>. They include:

Quizzes

There are quizzes at the end of each chapter that can be taken by the students. Students are encouraged to take these quizzes to test their general understanding of the materials presented in the corresponding chapter.

Solution to Odd-Numbered Practice Set

Solutions to all odd-numbered questions and problems are provided on the book website for the use of students.

Lab Assignments

The descriptions of lab assignments are also included in the student resources.

Applets

There are some applets for each chapter. Applets can either show the solution to some examples and problems or show some protocols in action. It is strongly recommended that students activate these applets.

Extra Materials

Students can also access the extra materials at the book website for further study.

How to Use the Book

The chapters in the book are organized to provide a great deal of flexibility. I suggest the following:

- ❑ Materials provided in Part I are essential for understanding the rest of the book.
- ❑ Part II (physical layer) is essential to understand the rest of the book, but the professor can skip this part if the students already have the background in engineering and the physical layer.
- ❑ Parts III to VI are based on the Internet model. They are required for understanding the use of the networking principle in the Internet.
- ❑ Part VII (QoS and Security) is related to all layers of the Internet mode. It can be partially or totally skipped if the students will be taking a course that covers these materials.

Website

The McGraw-Hill website contains much additional material, available at www.mhhe.com/forouzan. As students read through *Data Communications and Networking*, they can go online to take self-grading quizzes. They can also access lecture materials such as PowerPoint slides, and get additional review from animated figures from the book. Selected solutions are also available over the Web. The solutions to odd-numbered problems are provided to students, and instructors can use a password to access the complete set of solutions.

McGraw-Hill Create™



create

Craft your teaching resources to match the way you teach! With McGraw-Hill Create, www.mcgrawhillcreate.com, you can easily rearrange chapters, combine material from other content sources, and quickly upload content you have written like your course syllabus or teaching notes. Find the content you need in Create by searching through thousands of leading McGraw-Hill textbooks. Arrange your book to fit your teaching style. Create even allows you to personalize your book's appearance by selecting the cover and adding your name, school, and course information. Order a Create book and you'll receive a complimentary print review copy in 3–5 business days or a complimentary electronic review copy (eComp) via email in minutes. Go to www.mcgrawhillcreate.com today and register to experience how McGraw-Hill Create empowers you to teach *your* students *your* way.

Electronic Textbook Option

This text is offered through CourseSmart for both instructors and students. CourseSmart is an online resource where students can purchase the complete text online at almost half the cost of a traditional text. Purchasing the eTextbook allows students to take advantage of CourseSmart's web tools for learning, which include full text search, notes and highlighting, and email tools for sharing notes between classmates. To learn more about CourseSmart options, contact your sales representative or visit www.CourseSmart.com.

Acknowledgments

It is obvious that the development of a book of this scope needs the support of many people. I would like to acknowledge the contributions from peer reviewers to the development of the book. These reviewers are:

Tricha Anjali, Illinois Institute of Technology
Yoris A. Au, University of Texas at San Antonio
Randy J. Fortier, University of Windsor
Tirthankar Ghosh, Saint Cloud State University
Lawrence Hill, Rochester Institute of Technology
Ezzat Kirmani, Saint Cloud State University
Robert Koenke, University of Central Florida
Mike O'Dell, University of Texas at Arlington

Special thanks go to the staff of McGraw-Hill. Raghu Srinivasan, the publisher, proved how a proficient publisher can make the impossible, possible. Melinda Bilecki, the developmental editor, gave help whenever I needed it. Jane Mohr, the project manager, guided us through the production process with enormous enthusiasm. I also thank Dheeraj Chahal, full-service project manager, Brenda A. Rolwes, the cover designer, and Kathryn DiBernardo, the copy editor.

Behrouz A. Forouzan
Los Angeles, CA.
January 2012

TRADE MARK

Throughout the text we have used several trademarks. Rather than insert a trademark symbol with each mention of the trademark name, we acknowledge the trademarks here and state that they are used with no intention of infringing upon them. Other product names, trademarks, and registered trademarks are the property of their respective owners.

Overview

In the first part of the book, we discuss some general ideas related to both data communications and networking. This part lays the plan for the rest of the book. The part is made of two chapters that prepare the reader for the long journey ahead.

Chapter 1 Introduction

Chapter 2 Network Models

CHAPTER 1

Introduction

Data communications and networking have changed the way we do business and the way we live. Business decisions have to be made ever more quickly, and the decision makers require immediate access to accurate information. Why wait a week for that report from Europe to arrive by mail when it could appear almost instantaneously through computer networks? Businesses today rely on computer networks and internet-works.

Data communication and networking have found their way not only through business and personal communication, they have found many applications in political and social issues. People have found how to communicate with other people in the world to express their social and political opinions and problems. Communities in the world are not isolated anymore.

But before we ask how quickly we can get hooked up, we need to know how networks operate, what types of technologies are available, and which design best fills which set of needs.

This chapter paves the way for the rest of the book. It is divided into five sections.

- ❑ The first section introduces data communications and defines their components and the types of data exchanged. It also shows how different types of data are represented and how data is flowed through the network.
- ❑ The second section introduces networks and defines their criteria and structures. It introduces four different network topologies that are encountered throughout the book.
- ❑ The third section discusses different types of networks: LANs, WANs, and inter-networks (internets). It also introduces the Internet, the largest internet in the world. The concept of switching is also introduced in this section to show how small networks can be combined to create larger ones.
- ❑ The fourth section covers a brief history of the Internet. The section is divided into three eras: early history, the birth of the Internet, and the issues related to the Internet today. This section can be skipped if the reader is familiar with this history.
- ❑ The fifth section covers standards and standards organizations. The section covers Internet standards and Internet administration. We refer to these standards and organizations throughout the book.

1.1 DATA COMMUNICATIONS

When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance. The term **telecommunication**, which includes telephony, telegraphy, and television, means communication at a distance (*tele* is Greek for “far”). The word **data** refers to information presented in whatever form is agreed upon by the parties creating and using the data.

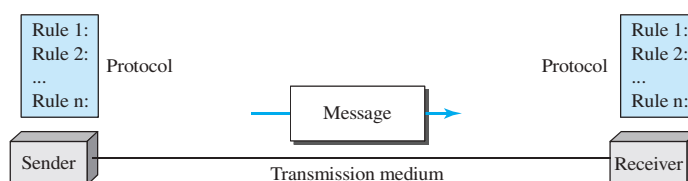
Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
2. **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.
4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

1.1.1 Components

A data communications system has five components (see Figure 1.1).

Figure 1.1 Five components of data communication



1. **Message.** The **message** is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. **Sender.** The **sender** is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The **receiver** is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. **Transmission medium.** The **transmission medium** is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.
5. **Protocol.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Representation

Information today comes in different forms such as text, numbers, images, audio, and video.

Text

In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a **code**, and the process of representing symbols is called coding. Today, the prevalent coding system is called **Unicode**, which uses 32 bits to represent a symbol or character used in any language in the world. The **American Standard Code for Information Interchange (ASCII)**, developed some decades ago in the United States, now constitutes the first 127 characters in Unicode and is also referred to as **Basic Latin**. Appendix A includes part of the Unicode.

Numbers

Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems.

Images

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the *resolution*. For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image.

After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only black-and-white dots (e.g., a chessboard), a 1-bit pattern is enough to represent a pixel.

If an image is not made of pure white and pure black pixels, we can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, we can use 2-bit patterns. A black pixel can be represented by 00, a dark gray pixel by 01, a light gray pixel by 10, and a white pixel by 11.

There are several methods to represent color images. One method is called **RGB**, so called because each color is made of a combination of three primary colors: *red*, *green*, and *blue*. The intensity of each color is measured, and a bit pattern is assigned to

it. Another method is called **YCM**, in which a color is made of a combination of three other primary colors: yellow, cyan, and *magenta*.

Audio

Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. We will learn more about audio in Chapter 26.

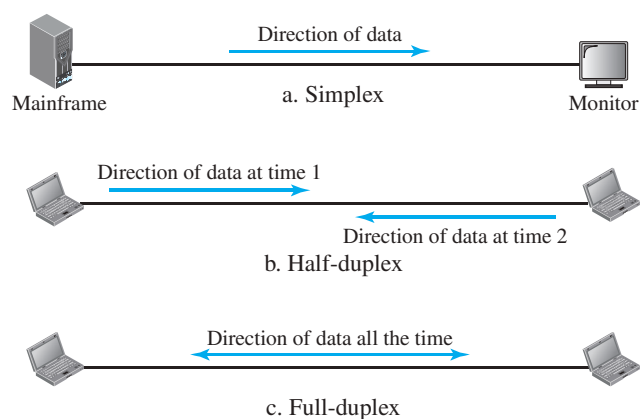
Video

Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion. We will learn more about video in Chapter 26.

1.1.3 Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure 1.2.

Figure 1.2 Data flow (simplex, half-duplex, and full-duplex)



Simplex

In **simplex mode**, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (see Figure 1.2a).

Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

Half-Duplex

In **half-duplex mode**, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (see Figure 1.2b).

The half-duplex mode is like a one-lane road with traffic allowed in both directions. When cars are traveling in one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

Full-Duplex

In **full-duplex mode** (also called *duplex*), both stations can transmit and receive simultaneously (see Figure 1.2c).

The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in one direction share the capacity of the link with signals going in the other direction. This sharing can occur in two ways: Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions.

One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

1.2 NETWORKS

A **network** is the interconnection of a set of devices capable of communication. In this definition, a device can be a **host** (or an *end system* as it is sometimes called) such as a large computer, desktop, laptop, workstation, cellular phone, or security system. A device in this definition can also be a **connecting device** such as a router, which connects the network to other networks, a switch, which connects devices together, a modem (modulator-demodulator), which changes the form of data, and so on. These devices in a network are connected using wired or wireless transmission media such as cable or air. When we connect two computers at home using a plug-and-play router, we have created a network, although very small.

1.2.1 Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

Performance is often evaluated by two networking metrics: **throughput** and **delay**. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability

In addition to accuracy of delivery, network **reliability** is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security

Network **security** issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

1.2.2 Physical Structures

Before discussing networks, we need to define some network attributes.

Type of Connection

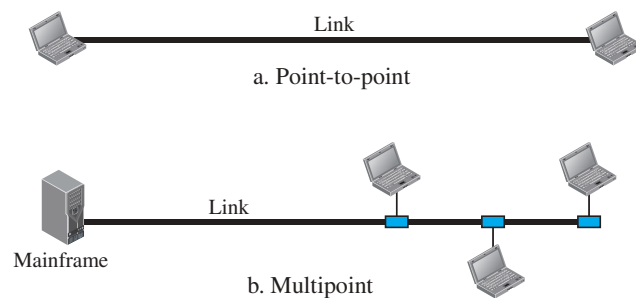
A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint.

Point-to-Point

A **point-to-point connection** provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible (see Figure 1.3a). When we change television channels by infrared remote control, we are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint

A **multipoint** (also called **multidrop**) **connection** is one in which more than two specific devices share a single link (see Figure 1.3b).

Figure 1.3 Types of connections: point-to-point and multipoint

In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.

Physical Topology

The term **physical topology** refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called **nodes**) to one another. There are four basic topologies possible: mesh, star, bus, and ring.

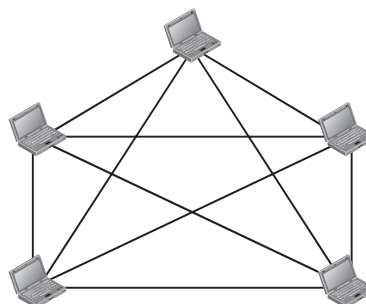
Mesh Topology

In a **mesh topology**, every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with n nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to $n - 1$ nodes, node 2 must be connected to $n - 1$ nodes, and finally node n must be connected to $n - 1$ nodes. We need $n(n - 1)$ physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need $n(n - 1) / 2$ duplex-mode links. To accommodate that many links, every device on the network must have $n - 1$ input/output (I/O) ports (see Figure 1.4) to be connected to the other $n - 1$ stations.

A mesh offers several advantages over other network topologies. First, the use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices. Second, a mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system. Third, there is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages. Finally, point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.

Figure 1.4 A fully connected mesh topology (five devices)

$n = 5$
10 links.

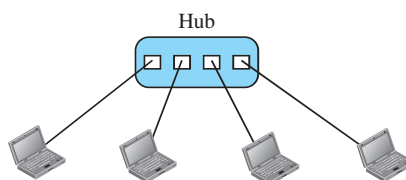


The main disadvantages of a mesh are related to the amount of cabling and the number of I/O ports required. First, because every device must be connected to every other device, installation and reconnection are difficult. Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate. Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive. For these reasons a mesh topology is usually implemented in a limited fashion, for example, as a backbone connecting the main computers of a hybrid network that can include several other topologies.

One practical example of a mesh topology is the connection of telephone regional offices in which each regional office needs to be connected to every other regional office.

Star Topology

In a **star topology**, each device has a dedicated point-to-point link only to a central controller, usually called a **hub**. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device (see Figure 1.5).

Figure 1.5 A star topology connecting four stations

A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and

additions, moves, and deletions involve only one connection: between that device and the hub.

Other advantages include robustness. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

One big disadvantage of a star topology is the dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.

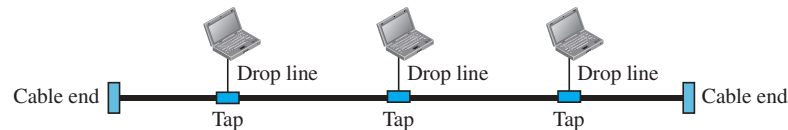
Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).

The star topology is used in local-area networks (LANs), as we will see in Chapter 13. High-speed LANs often use a star topology with a central hub.

Bus Topology

The preceding examples all describe point-to-point connections. A **bus topology**, on the other hand, is multipoint. One long cable acts as a **backbone** to link all the devices in a network (see Figure 1.6).

Figure 1.6 A bus topology connecting three stations



Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason there is a limit on the number of taps a bus can support and on the distance between those taps.

Advantages of a bus topology include ease of installation. Backbone cable can be laid along the most efficient path, then connected to the nodes by drop lines of various lengths. In this way, a bus uses less cabling than mesh or star topologies. In a star, for example, four network devices in the same room require four lengths of cable reaching all the way to the hub. In a bus, this redundancy is eliminated. Only the backbone cable stretches through the entire facility. Each drop line has to reach only as far as the nearest point on the backbone.

Disadvantages include difficult reconnection and fault isolation. A bus is usually designed to be optimally efficient at installation. It can therefore be difficult to add new devices. Signal reflection at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to a given

length of cable. Adding new devices may therefore require modification or replacement of the backbone.

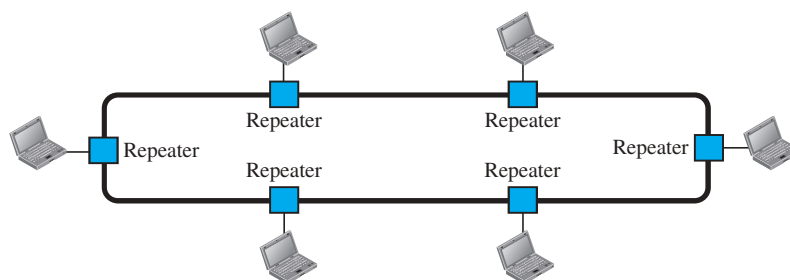
In addition, a fault or break in the bus cable stops all transmission, even between devices on the same side of the problem. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

Bus topology was the one of the first topologies used in the design of early local-area networks. Traditional Ethernet LANs can use a bus topology, but they are less popular now for reasons we will discuss in Chapter 13.

Ring Topology

In a **ring topology**, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along (see Figure 1.7).

Figure 1.7 A ring topology connecting six stations



A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). In addition, fault isolation is simplified. Generally, in a ring a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

However, unidirectional traffic can be a disadvantage. In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break.

Ring topology was prevalent when IBM introduced its local-area network, Token Ring. Today, the need for higher-speed LANs has made this topology less popular.

1.3 NETWORK TYPES

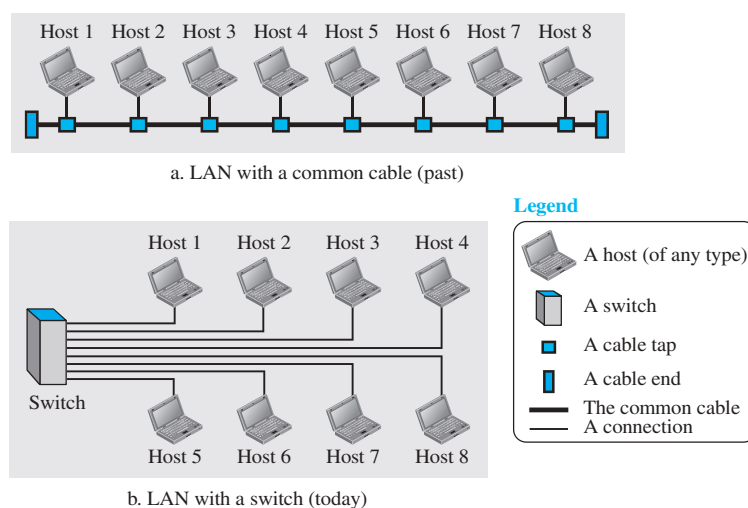
After defining networks in the previous section and discussing their physical structures, we need to discuss different types of networks we encounter in the world today. The criteria of distinguishing one type of network from another is difficult and sometimes confusing. We use a few criteria such as size, geographical coverage, and ownership to make this distinction. After discussing two types of networks, LANs and WANs, we define switching, which is used to connect networks to form an internetwork (a network of networks).

1.3.1 Local Area Network

A **local area network (LAN)** is usually privately owned and connects some hosts in a single office, building, or campus. Depending on the needs of an organization, a LAN can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video devices. Each host in a LAN has an identifier, an address, that uniquely defines the host in the LAN. A packet sent by a host to another host carries both the source host's and the destination host's addresses.

In the past, all hosts in a network were connected through a common cable, which meant that a packet sent from one host to another was received by all hosts. The intended recipient kept the packet; the others dropped the packet. Today, most LANs use a smart connecting switch, which is able to recognize the destination address of the packet and guide the packet to its destination without sending it to all other hosts. The switch alleviates the traffic in the LAN and allows more than one pair to communicate with each other at the same time if there is no common source and destination among them. Note that the above definition of a LAN does not define the minimum or maximum number of hosts in a LAN. Figure 1.8 shows a LAN using either a common cable or a switch.

Figure 1.8 An isolated LAN in the past and today



LANs are discussed in more detail in Part III of the book.

When LANs were used in isolation (which is rare today), they were designed to allow resources to be shared between the hosts. As we will see shortly, LANs today are connected to each other and to WANs (discussed next) to create communication at a wider level.

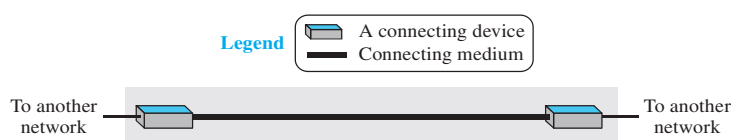
1.3.2 Wide Area Network

A **wide area network (WAN)** is also an interconnection of devices capable of communication. However, there are some differences between a LAN and a WAN. A LAN is normally limited in size, spanning an office, a building, or a campus; a WAN has a wider geographical span, spanning a town, a state, a country, or even the world. A LAN interconnects hosts; a WAN interconnects connecting devices such as switches, routers, or modems. A LAN is normally privately owned by the organization that uses it; a WAN is normally created and run by communication companies and leased by an organization that uses it. We see two distinct examples of WANs today: point-to-point WANs and switched WANs.

Point-to-Point WAN

A point-to-point WAN is a network that connects two communicating devices through a transmission media (cable or air). We will see examples of these WANs when we discuss how to connect the networks to one another. Figure 1.9 shows an example of a point-to-point WAN.

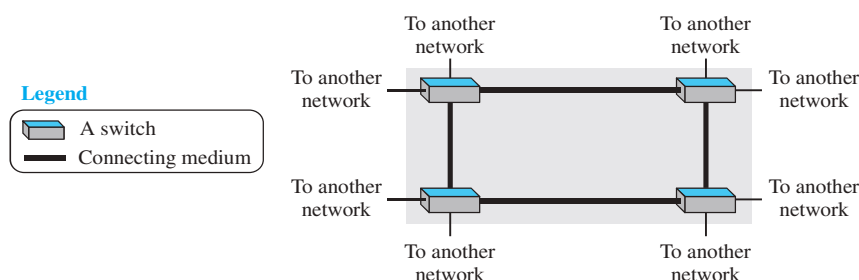
Figure 1.9 A point-to-point WAN



Switched WAN

A switched WAN is a network with more than two ends. A switched WAN, as we will see shortly, is used in the backbone of global communication today. We can say that a switched WAN is a combination of several point-to-point WANs that are connected by switches. Figure 1.10 shows an example of a switched WAN.

Figure 1.10 A switched WAN

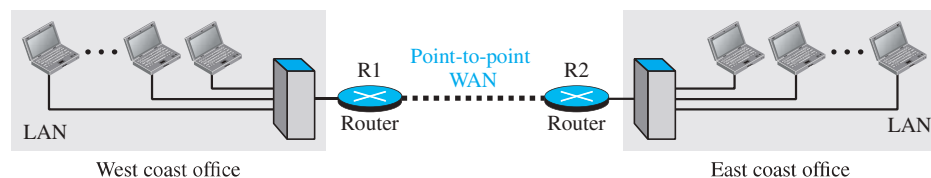


WANs are discussed in more detail in Part II of the book.

Internetwork

Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an **internetwork**, or **internet**. As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. Each office has a LAN that allows all employees in the office to communicate with each other. To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs. Now the company has an internetwork, or a private internet (with lowercase *i*). Communication between offices is now possible. Figure 1.11 shows this internet.

Figure 1.11 An internetwork made of two LANs and one point-to-point WAN



When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination. On the other hand, when a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination.

Figure 1.12 (see next page) shows another internet with several LANs and WANs connected. One of the WANs is a switched WAN with four switches.

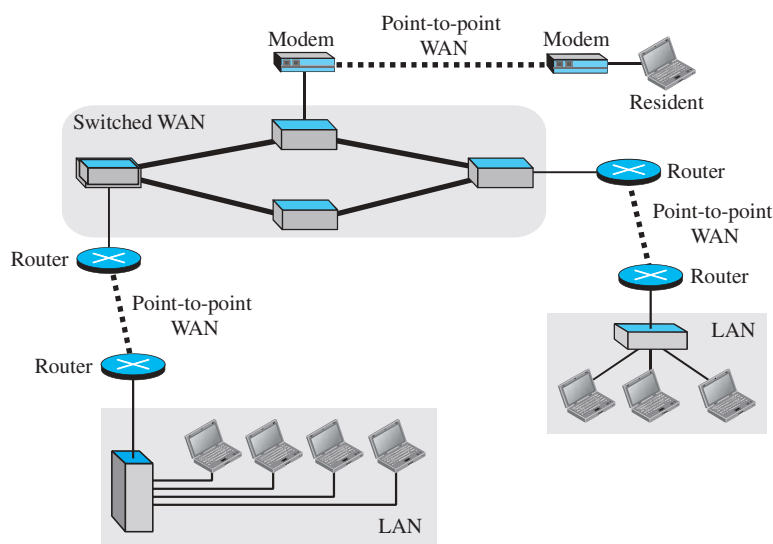
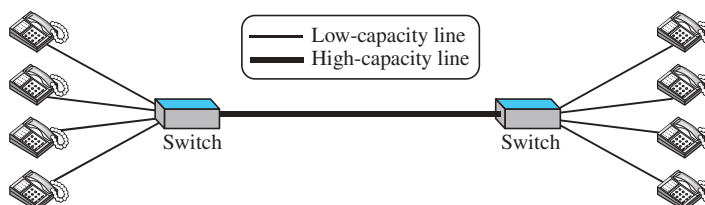
1.3.3 Switching

An internet is a **switched network** in which a switch connects at least two links together. A switch needs to forward data from a network to another network when required. The two most common types of switched networks are circuit-switched and packet-switched networks. We discuss both next.

Circuit-Switched Network

In a **circuit-switched network**, a dedicated connection, called a circuit, is always available between the two end systems; the switch can only make it active or inactive. Figure 1.13 shows a very simple switched network that connects four telephones to each end. We have used telephone sets instead of computers as an end system because circuit switching was very common in telephone networks in the past, although part of the telephone network today is a packet-switched network.

In Figure 1.13, the four telephones at each side are connected to a switch. The switch connects a telephone set at one side to a telephone set at the other side. The thick

Figure 1.12 *A heterogeneous network made of four WANs and three LANs***Figure 1.13** *A circuit-switched network*

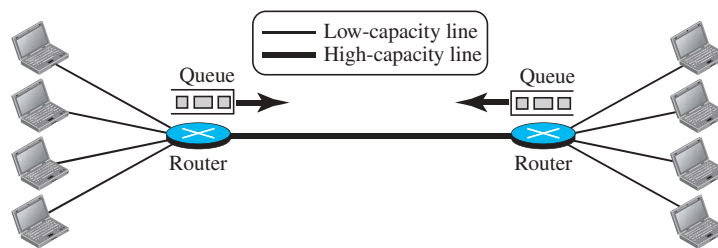
line connecting two switches is a high-capacity communication line that can handle four voice communications at the same time; the capacity can be shared between all pairs of telephone sets. The switches used in this example have forwarding tasks but no storing capability.

Let us look at two cases. In the first case, all telephone sets are busy; four people at one site are talking with four people at the other site; the capacity of the thick line is fully used. In the second case, only one telephone set at one side is connected to a telephone set at the other side; only one-fourth of the capacity of the thick line is used. This means that a circuit-switched network is efficient only when it is working at its full capacity; most of the time, it is inefficient because it is working at partial capacity. The reason that we need to make the capacity of the thick line four times the capacity of each voice line is that we do not want communication to fail when all telephone sets at one side want to be connected with all telephone sets at the other side.

Packet-Switched Network

In a computer network, the communication between the two ends is done in blocks of data called **packets**. In other words, instead of the continuous communication we see between two telephone sets when they are being used, we see the exchange of individual data packets between the two computers. This allows us to make the switches function for both storing and forwarding because a packet is an independent entity that can be stored and sent later. Figure 1.14 shows a small packet-switched network that connects four computers at one site to four computers at the other site.

Figure 1.14 A packet-switched network



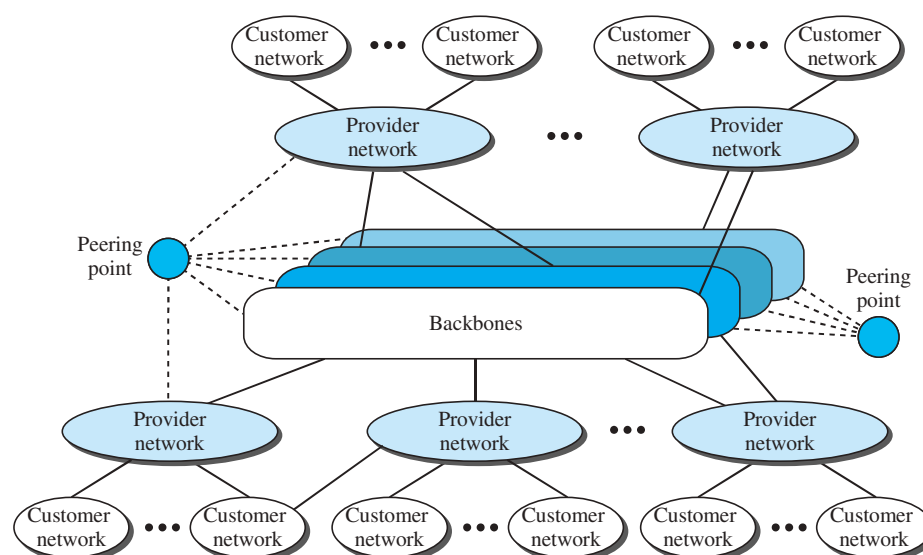
A router in a packet-switched network has a queue that can store and forward the packet. Now assume that the capacity of the thick line is only twice the capacity of the data line connecting the computers to the routers. If only two computers (one at each site) need to communicate with each other, there is no waiting for the packets. However, if packets arrive at one router when the thick line is already working at its full capacity, the packets should be stored and forwarded in the order they arrived. The two simple examples show that a packet-switched network is more efficient than a circuit-switched network, but the packets may encounter some delays.

In this book, we mostly discuss packet-switched networks. In Chapter 18, we discuss packet-switched networks in more detail and discuss the performance of these networks.

1.3.4 The Internet

As we discussed before, an internet (note the lowercase *i*) is two or more networks that can communicate with each other. The most notable internet is called the **Internet** (uppercase *I*), and is composed of thousands of interconnected networks. Figure 1.15 shows a conceptual (not geographical) view of the Internet.

The figure shows the Internet as several backbones, provider networks, and customer networks. At the top level, the *backbones* are large networks owned by some communication companies such as Sprint, Verizon (MCI), AT&T, and NTT. The backbone networks are connected through some complex switching systems, called *peering points*. At the second level, there are smaller networks, called *provider networks*, that use the services of the backbones for a fee. The provider networks are connected to backbones and sometimes to other provider networks. The *customer networks* are

Figure 1.15 *The Internet today*

networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services.

Backbones and provider networks are also called **Internet Service Providers (ISPs)**. The backbones are often referred to as *international ISPs*; the provider networks are often referred to as *national* or *regional ISPs*.

1.3.5 Accessing the Internet

The Internet today is an internetwork that allows any user to become part of it. The user, however, needs to be physically connected to an ISP. The physical connection is normally done through a point-to-point WAN. In this section, we briefly describe how this can happen, but we postpone the technical details of the connection until Chapters 14 and 16.

Using Telephone Networks

Today most residences and small businesses have telephone service, which means they are connected to a telephone network. Since most telephone networks have already connected themselves to the Internet, one option for residences and small businesses to connect to the Internet is to change the voice line between the residence or business and the telephone center to a point-to-point WAN. This can be done in two ways.

- ❑ **Dial-up service.** The first solution is to add to the telephone line a modem that converts data to voice. The software installed on the computer dials the ISP and imitates making a telephone connection. Unfortunately, the dial-up service is

very slow, and when the line is used for Internet connection, it cannot be used for telephone (voice) connection. It is only useful for small residences. We discuss dial-up service in Chapter 14.

- **DSL Service.** Since the advent of the Internet, some telephone companies have upgraded their telephone lines to provide higher speed Internet services to residences or small businesses. The DSL service also allows the line to be used simultaneously for voice and data communication. We discuss DSL in Chapter 14.

Using Cable Networks

More and more residents over the last two decades have begun using cable TV services instead of antennas to receive TV broadcasting. The cable companies have been upgrading their cable networks and connecting to the Internet. A residence or a small business can be connected to the Internet by using this service. It provides a higher speed connection, but the speed varies depending on the number of neighbors that use the same cable. We discuss the cable networks in Chapter 14.

Using Wireless Networks

Wireless connectivity has recently become increasingly popular. A household or a small business can use a combination of wireless and wired connections to access the Internet. With the growing wireless WAN access, a household or a small business can be connected to the Internet through a wireless WAN. We discuss wireless access in Chapter 16.

Direct Connection to the Internet

A large organization or a large corporation can itself become a local ISP and be connected to the Internet. This can be done if the organization or the corporation leases a high-speed WAN from a carrier provider and connects itself to a regional ISP. For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet.

1.4 INTERNET HISTORY

Now that we have given an overview of the Internet, let us give a brief history of the Internet. This brief history makes it clear how the Internet has evolved from a private network to a global one in less than 40 years.

1.4.1 Early History

There were some communication networks, such as telegraph and telephone networks, before 1960. These networks were suitable for constant-rate communication at that time, which means that after a connection was made between two users, the encoded message (telegraphy) or voice (telephony) could be exchanged. A computer network, on the other hand, should be able to handle *bursty* data, which means data received at variable rates at different times. The world needed to wait for the packet-switched network to be invented.

Birth of Packet-Switched Networks

The theory of packet switching for bursty traffic was first presented by Leonard Kleinrock in 1961 at MIT. At the same time, two other researchers, Paul Baran at Rand Institute and Donald Davies at National Physical Laboratory in England, published some papers about packet-switched networks.

ARPANET

In the mid-1960s, mainframe computers in research organizations were stand-alone devices. Computers from different manufacturers were unable to communicate with one another. The **Advanced Research Projects Agency (ARPA)** in the Department of Defense (DOD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for the **Advanced Research Projects Agency Network (ARPANET)**, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *interface message processor* (IMP). The IMPs, in turn, would be connected to each other. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

1.4.2 Birth of the Internet

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetworking Project*. They wanted to link dissimilar networks so that a host on one network could communicate with a host on another. There were many problems to overcome: diverse packet sizes, diverse interfaces, and diverse transmission rates, as well as differing reliability requirements. Cerf and Kahn devised the idea of a device called a *gateway* to serve as the intermediary hardware to transfer data from one network to another.

TCP/IP

Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of data. This was a new version of NCP. This paper on transmission control protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. A radical idea was the transfer of responsibility for error correction from the IMP to the host machine. This ARPA Internet now became the focus of the communication effort. Around this time, responsibility for the ARPANET was handed over to the Defense Communication Agency (DCA).

In October 1977, an internet consisting of three different networks (ARPANET, packet radio, and packet satellite) was successfully demonstrated. Communication between networks was now possible.

Shortly thereafter, authorities made a decision to split TCP into two protocols: **Transmission Control Protocol (TCP)** and **Internet Protocol (IP)**. IP would handle datagram routing while TCP would be responsible for higher level functions such as segmentation, reassembly, and error detection. The new combination became known as TCP/IP.

In 1981, under a Defence Department contract, UC Berkeley modified the UNIX operating system to include TCP/IP. This inclusion of network software along with a popular operating system did much for the popularity of internetworking. The open (non-manufacturer-specific) implementation of the Berkeley UNIX gave every manufacturer a working code base on which they could build their products.

In 1983, authorities abolished the original ARPANET protocols, and TCP/IP became the official protocol for the ARPANET. Those who wanted to use the Internet to access a computer on a different network had to be running TCP/IP.

MILNET

In 1983, ARPANET split into two networks: **Military Network (MILNET)** for military users and ARPANET for nonmilitary users.

CSNET

Another milestone in Internet history was the creation of CSNET in 1981. **Computer Science Network (CSNET)** was a network sponsored by the National Science Foundation (NSF). The network was conceived by universities that were ineligible to join ARPANET due to an absence of ties to the Department of Defense. CSNET was a less expensive network; there were no redundant links and the transmission rate was slower.

By the mid-1980s, most U.S. universities with computer science departments were part of CSNET. Other institutions and companies were also forming their own networks and using TCP/IP to interconnect. The term *Internet*, originally associated with government-funded connected networks, now referred to the connected networks using TCP/IP protocols.

NSFNET

With the success of CSNET, the NSF in 1986 sponsored the **National Science Foundation Network (NSFNET)**, a backbone that connected five supercomputer centers located throughout the United States. Community networks were allowed access to this backbone, a T-1 line (see Chapter 6) with a 1.544-Mbps data rate, thus providing connectivity throughout the United States. In 1990, ARPANET was officially retired and replaced by NSFNET. In 1995, NSFNET reverted back to its original concept of a research network.

ANSNET

In 1991, the U.S. government decided that NSFNET was not capable of supporting the rapidly increasing Internet traffic. Three companies, IBM, Merit, and Verizon, filled the void by forming a nonprofit organization called Advanced Network & Services (ANS) to build a new, high-speed Internet backbone called **Advanced Network Services Network (ANSNET)**.

1.4.3 Internet Today

Today, we witness a rapid growth both in the infrastructure and new applications. The Internet today is a set of peer networks that provide services to the whole world. What has made the Internet so popular is the invention of new applications.

World Wide Web

The 1990s saw the explosion of Internet applications due to the emergence of the World Wide Web (WWW). The Web was invented at CERN by Tim Berners-Lee. This invention has added the commercial applications to the Internet.

Multimedia

Recent developments in the multimedia applications such as voice over IP (telephony), video over IP (Skype), view sharing (YouTube), and television over IP (PPLive) has increased the number of users and the amount of time each user spends on the network. We discuss multimedia in Chapter 28.

Peer-to-Peer Applications

Peer-to-peer networking is also a new area of communication with a lot of potential. We introduce some peer-to-peer applications in Chapter 29.

1.5 STANDARDS AND ADMINISTRATION

In the discussion of the Internet and its protocol, we often see a reference to a standard or an administration entity. In this section, we introduce these standards and administration entities for those readers that are not familiar with them; the section can be skipped if the reader is familiar with them.

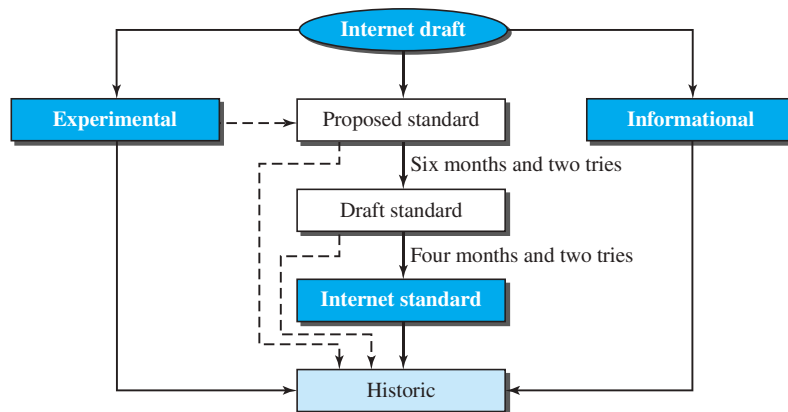
1.5.1 Internet Standards

An **Internet standard** is a thoroughly tested specification that is useful to and adhered to by those who work with the Internet. It is a formalized regulation that must be followed. There is a strict procedure by which a specification attains Internet standard status. A specification begins as an Internet draft. An **Internet draft** is a working document (a work in progress) with no official status and a six-month lifetime. Upon recommendation from the Internet authorities, a draft may be published as a **Request for Comment (RFC)**. Each RFC is edited, assigned a number, and made available to all interested parties. RFCs go through maturity levels and are categorized according to their requirement level.

Maturity Levels

An RFC, during its lifetime, falls into one of six *maturity levels*: proposed standard, draft standard, Internet standard, historic, experimental, and informational (see Figure 1.16).

- **Proposed Standard.** A proposed standard is a specification that is stable, well understood, and of sufficient interest to the Internet community. At this level, the specification is usually tested and implemented by several different groups.

Figure 1.16 *Maturity levels of an RFC*

- ❑ **Draft Standard.** A proposed standard is elevated to draft standard status after at least two successful independent and interoperable implementations. Barring difficulties, a draft standard, with modifications if specific problems are encountered, normally becomes an Internet standard.
- ❑ **Internet Standard.** A draft standard reaches Internet standard status after demonstrations of successful implementation.
- ❑ **Historic.** The historic RFCs are significant from a historical perspective. They either have been superseded by later specifications or have never passed the necessary maturity levels to become an Internet standard.
- ❑ **Experimental.** An RFC classified as experimental describes work related to an experimental situation that does not affect the operation of the Internet. Such an RFC should not be implemented in any functional Internet service.
- ❑ **Informational.** An RFC classified as informational contains general, historical, or tutorial information related to the Internet. It is usually written by someone in a non-Internet organization, such as a vendor.

Requirement Levels

RFCs are classified into five *requirement levels*: required, recommended, elective, limited use, and not recommended.

- ❑ **Required.** An RFC is labeled *required* if it must be implemented by all Internet systems to achieve minimum conformance. For example, IP and ICMP (Chapter 19) are required protocols.
- ❑ **Recommended.** An RFC labeled recommended is not required for minimum conformance; it is recommended because of its usefulness. For example, FTP (Chapter 26) and TELNET (Chapter 26) are recommended protocols.
- ❑ **Elective.** An RFC labeled elective is not required and not recommended. However, a system can use it for its own benefit.

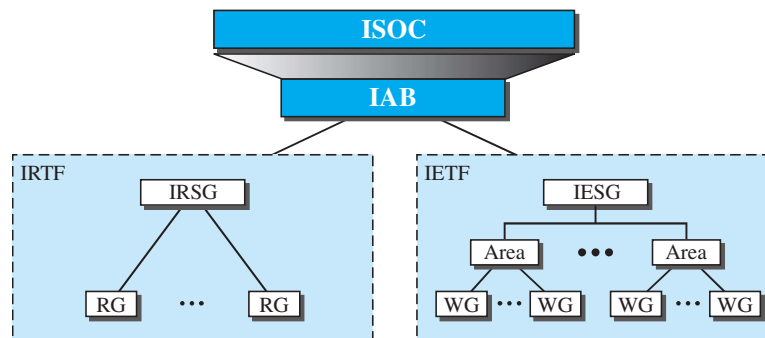
- ❑ **Limited Use.** An RFC labeled limited use should be used only in limited situations. Most of the experimental RFCs fall under this category.
- ❑ **Not Recommended.** An RFC labeled not recommended is inappropriate for general use. Normally a historic (deprecated) RFC may fall under this category.

RFCs can be found at <http://www.rfc-editor.org>.

1.5.2 Internet Administration

The Internet, with its roots primarily in the research domain, has evolved and gained a broader user base with significant commercial activity. Various groups that coordinate Internet issues have guided this growth and development. Appendix G gives the addresses, e-mail addresses, and telephone numbers for some of these groups. Figure 1.17 shows the general organization of Internet administration.

Figure 1.17 Internet administration



ISOC

The **Internet Society (ISOC)** is an international, nonprofit organization formed in 1992 to provide support for the Internet standards process. ISOC accomplishes this through maintaining and supporting other Internet administrative bodies such as IAB, IETF, IRTF, and IANA (see the following sections). ISOC also promotes research and other scholarly activities relating to the Internet.

IAB

The **Internet Architecture Board (IAB)** is the technical advisor to the ISOC. The main purposes of the IAB are to oversee the continuing development of the TCP/IP Protocol Suite and to serve in a technical advisory capacity to research members of the Internet community. IAB accomplishes this through its two primary components, the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF). Another responsibility of the IAB is the editorial management of the RFCs, described

earlier. IAB is also the external liaison between the Internet and other standards organizations and forums.

IETF

The **Internet Engineering Task Force (IETF)** is a forum of working groups managed by the Internet Engineering Steering Group (IESG). IETF is responsible for identifying operational problems and proposing solutions to these problems. IETF also develops and reviews specifications intended as Internet standards. The working groups are collected into areas, and each area concentrates on a specific topic. Currently nine areas have been defined. The areas include applications, protocols, routing, network management next generation (IPng), and security.

IRTF

The **Internet Research Task Force (IRTF)** is a forum of working groups managed by the Internet Research Steering Group (IRSG). IRTF focuses on long-term research topics related to Internet protocols, applications, architecture, and technology.

1.6 END-CHAPTER MATERIALS

1.6.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books. The items enclosed in brackets [. . .] refer to the reference list at the end of the book.

Books

The introductory materials covered in this chapter can be found in [Sta04] and [PD03]. [Tan03] also discusses standardization.

1.6.2 Key Terms

Advanced Network Services Network (ANSNET)	full-duplex mode
Advanced Research Projects Agency (ARPA)	half-duplex mode
Advanced Research Projects Agency Network (ARPANET)	hub
American Standard Code for Information Interchange (ASCII)	image
audio	internet
backbone	Internet
Basic Latin	Internet Architecture Board (IAB)
bus topology	Internet draft
circuit-switched network	Internet Engineering Task Force (IETF)
code	Internet Research Task Force (IRTF)
Computer Science Network (CSNET)	Internet Service Provider (ISP)
data	Internet Society (ISOC)
data communications	Internet standard
delay	internetwork
	local area network (LAN)
	mesh topology
	message

Military Network (MILNET)	ring topology
multipoint or multidrop connection	simplex mode
National Science Foundation Network (NSFNET)	star topology
network	switched network
node	TCP/IP protocol suite
packet	telecommunication
packet-switched network	throughput
performance	Transmission Control Protocol/ Internet Protocol (TCP/IP)
physical topology	transmission medium
point-to-point connection	Unicode
protocol	video
Request for Comment (RFC)	wide area network (WAN)
RGB	YCM

1.6.3 Summary

Data communications are the transfer of data from one device to another via some form of transmission medium. A data communications system must transmit data to the correct destination in an accurate and timely manner. The five components that make up a data communications system are the message, sender, receiver, medium, and protocol. Text, numbers, images, audio, and video are different forms of information. Data flow between two devices can occur in one of three ways: simplex, half-duplex, or full-duplex.

A network is a set of communication devices connected by media links. In a point-to-point connection, two and only two devices are connected by a dedicated link. In a multipoint connection, three or more devices share a link. Topology refers to the physical or logical arrangement of a network. Devices may be arranged in a mesh, star, bus, or ring topology.

A network can be categorized as a local area network or a wide area network. A LAN is a data communication system within a building, plant, or campus, or between nearby buildings. A WAN is a data communication system spanning states, countries, or the whole world. An internet is a network of networks. The Internet is a collection of many separate networks.

The Internet history started with the theory of packet switching for bursty traffic. The history continued when The ARPA was interested in finding a way to connect computers so that the researchers they funded could share their findings, resulting in the creation of ARPANET. The Internet was born when Cerf and Kahn devised the idea of a device called a *gateway* to serve as the intermediary hardware to transfer data from one network to another. The TCP/IP protocol suite paved the way for creation of today's Internet. The invention of WWW, the use of multimedia, and peer-to-peer communication helps the growth of the Internet.

An Internet standard is a thoroughly tested specification. An Internet draft is a working document with no official status and a six-month lifetime. A draft may be published as a Request for Comment (RFC). RFCs go through maturity levels and are categorized according to their requirement level. The Internet administration has

evolved with the Internet. ISOC promotes research and activities. IAB is the technical advisor to the ISOC. IETF is a forum of working groups responsible for operational problems. IRTF is a forum of working groups focusing on long-term research topics.

1.7 PRACTICE SET

1.7.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

1.7.2 Questions

- Q1-1.** Identify the five components of a data communications system.
- Q1-2.** What are the three criteria necessary for an effective and efficient network?
- Q1-3.** What are the advantages of a multipoint connection over a point-to-point one?
- Q1-4.** What are the two types of line configuration?
- Q1-5.** Categorize the four basic topologies in terms of line configuration.
- Q1-6.** What is the difference between half-duplex and full-duplex transmission modes?
- Q1-7.** Name the four basic network topologies, and cite an advantage of each type.
- Q1-8.** For n devices in a network, what is the number of cable links required for a mesh, ring, bus, and star topology?
- Q1-9.** What are some of the factors that determine whether a communication system is a LAN or WAN?
- Q1-10.** What is an internet? What is the Internet?
- Q1-11.** Why are protocols needed?
- Q1-12.** In a LAN with a link-layer switch (Figure 1.8b), Host 1 wants to send a message to Host 3. Since communication is through the link-layer switch, does the switch need to have an address? Explain.
- Q1-13.** How many point-to-point WANs are needed to connect n LANs if each LAN should be able to directly communicate with any other LAN?
- Q1-14.** When we use local telephones to talk to a friend, are we using a circuit-switched network or a packet-switched network?
- Q1-15.** When a resident uses a dial-up or DLS service to connect to the Internet, what is the role of the telephone company?
- Q1-16.** What is the first principle we discussed in this chapter for protocol layering that needs to be followed to make the communication bidirectional?
- Q1-17.** Explain the difference between an Internet draft and a proposed standard.
- Q1-18.** Explain the difference between a required RFC and a recommended RFC.
- Q1-19.** Explain the difference between the duties of the IETF and IRTF.

1.7.3 Problems

- P1-1.** What is the maximum number of characters or symbols that can be represented by Unicode?
- P1-2.** A color image uses 16 bits to represent a pixel. What is the maximum number of different colors that can be represented?
- P1-3.** Assume six devices are arranged in a mesh topology. How many cables are needed? How many ports are needed for each device?
- P1-4.** For each of the following four networks, discuss the consequences if a connection fails.
 - a.** Five devices arranged in a mesh topology
 - b.** Five devices arranged in a star topology (not counting the hub)
 - c.** Five devices arranged in a bus topology
 - d.** Five devices arranged in a ring topology
- P1-5.** We have two computers connected by an Ethernet hub at home. Is this a LAN or a WAN? Explain the reason.
- P1-6.** In the ring topology in Figure 1.7, what happens if one of the stations is unplugged?
- P1-7.** In the bus topology in Figure 1.6, what happens if one of the stations is unplugged?
- P1-8.** Performance is inversely related to delay. When we use the Internet, which of the following applications are more sensitive to delay?
 - a.** Sending an e-mail
 - b.** Copying a file
 - c.** Surfing the Internet
- P1-9.** When a party makes a local telephone call to another party, is this a point-to-point or multipoint connection? Explain the answer.
- P1-10.** Compare the telephone network and the Internet. What are the similarities? What are the differences?

1.8 SIMULATION EXPERIMENTS

1.8.1 Applets

One of the ways to show the network protocols in action or visually see the solution to some examples is through the use of interactive animation. We have created some Java applets to show some of the main concepts discussed in this chapter. It is strongly recommended that the students activate these applets on the book website and carefully examine the protocols in action. However, note that applets have been created only for some chapters, not all (see the book website).

1.8.2 Lab Assignments

Experiments with networks and network equipment can be done using at least two methods. In the first method, we can create an isolated networking laboratory and use

networking hardware and software to simulate the topics discussed in each chapter. We can create an internet and send and receive packets from any host to another. The flow of packets can be observed and the performance can be measured. Although the first method is more effective and more instructional, it is expensive to implement and not all institutions are ready to invest in such an exclusive laboratory.

In the second method, we can use the Internet, the largest network in the world, as our virtual laboratory. We can send and receive packets using the Internet. The existence of some free-downloadable software allows us to capture and examine the packets exchanged. We can analyze the packets to see how theoretical aspects of networking are put into action. Although the second method may not be as effective as the first method, in that we cannot control and change the packet routes to see how the Internet behaves, the method is much cheaper to implement. It does not need a physical networking lab; it can be implemented using our desktop or laptop. The required software is also free to download.

There are many programs and utilities available for Windows and UNIX operating systems that allow us to sniff, capture, trace, and analyze packets that are exchanged between our computer and the Internet. Some of these, such as *Wireshark* and *Ping-Plotter*, have graphical user interface (GUI); others, such as *tracert*, *nslookup*, *dig*, *ipconfig*, and *ifconfig*, are network administration command-line utilities. Any of these programs and utilities can be a valuable debugging tool for network administrators and educational tool for computer network students.

In this book, we mostly use Wireshark for lab assignments, although we occasionally use other tools. It captures live packet data from a network interface and displays them with detailed protocol information. Wireshark, however, is a passive analyzer. It only “measures” things from the network without manipulating them; it doesn’t send packets on the network or perform other active operations. Wireshark is not an intrusion detection tool either. It does not give warning about any network intrusion. It, nevertheless, can help network administrators or network security engineers to figure out what is going on inside a network and to troubleshoot network problems. In addition to being an indispensable tool for network administrators and security engineers, Wireshark is a valuable tool for protocol developers, who may use it to debug protocol implementations, and a great educational tool for computer networking students who can use it to see details of protocol operations in real time. However, note that we can use lab assignments only with a few chapters.

Lab1-1. In this lab assignment we learn how to download and install Wireshark. The instructions for downloading and installing the software are posted on the book website in the lab section for Chapter 1. In this document, we also discuss the general idea behind the software, the format of its window, and how to use it. The full study of this lab prepares the student to use Wireshark in the lab assignments for other chapters.

CHAPTER 2

Network Models

The second chapter is a preparation for the rest of the book. The next five parts of the book is devoted to one of the layers in the TCP/IP protocol suite. In this chapter, we first discuss the idea of network models in general and the TCP/IP protocol suite in particular.

Two models have been devised to define computer network operations: the TCP/IP protocol suite and the OSI model. In this chapter, we first discuss a general subject, protocol layering, which is used in both models. We then concentrate on the TCP/IP protocol suite, on which the book is based. The OSI model is briefly discuss for comparison with the TCP/IP protocol suite.

- The first section introduces the concept of protocol layering using two scenarios. The section also discusses the two principles upon which the protocol layering is based. The first principle dictates that each layer needs to have two opposite tasks. The second principle dictates that the corresponding layers should be identical. The section ends with a brief discussion of logical connection between two identical layers in protocol layering. Throughout the book, we need to distinguish between logical and physical connections.
- The second section discusses the five layers of the TCP/IP protocol suite. We show how packets in each of the five layers (physical, data-link, network, transport, and application) are named. We also mention the addressing mechanism used in each layer. Each layer of the TCP/IP protocol suite is a subject of a part of the book. In other words, each layer is discussed in several chapters; this section is just an introduction and preparation.
- The third section gives a brief discussion of the OSI model. This model was never implemented in practice, but a brief discussion of the model and its comparison with the TCP/IP protocol suite may be useful to better understand the TCP/IP protocol suite. In this section we also give a brief reason for the OSI model's lack of success.

2.1 PROTOCOL LAYERING

We defined the term *protocol* in Chapter 1. In data communication and networking, a protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively. When communication is simple, we may need only one simple protocol; when the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or **protocol layering**.

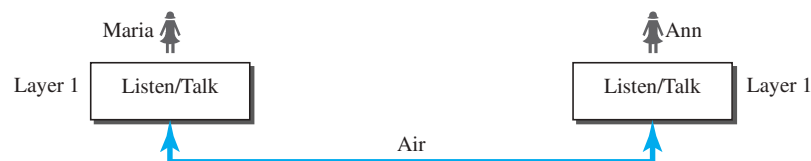
2.1.1 Scenarios

Let us develop two simple scenarios to better understand the need for protocol layering.

First Scenario

In the first scenario, communication is so simple that it can occur in only one layer. Assume Maria and Ann are neighbors with a lot of common ideas. Communication between Maria and Ann takes place in one layer, face to face, in the same language, as shown in Figure 2.1.

Figure 2.1 A single-layer protocol



Even in this simple scenario, we can see that a set of rules needs to be followed. First, Maria and Ann know that they should greet each other when they meet. Second, they know that they should confine their vocabulary to the level of their friendship. Third, each party knows that she should refrain from speaking when the other party is speaking. Fourth, each party knows that the conversation should be a dialog, not a monolog: both should have the opportunity to talk about the issue. Fifth, they should exchange some nice words when they leave.

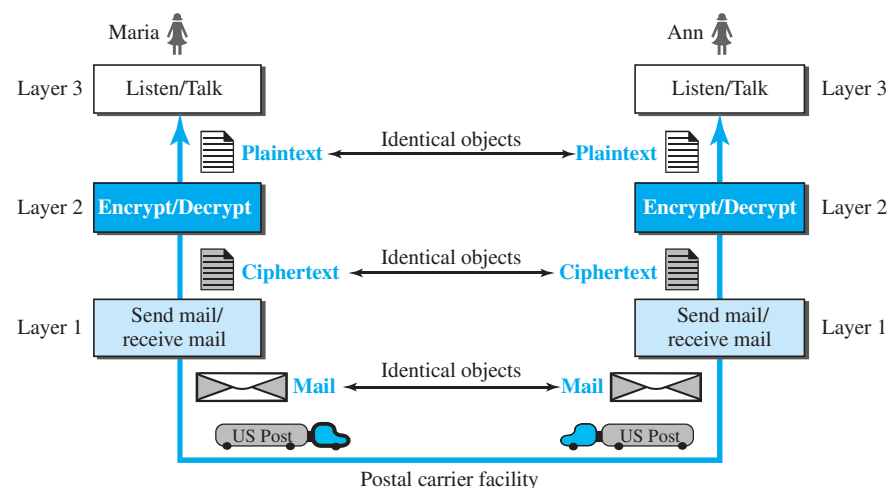
We can see that the protocol used by Maria and Ann is different from the communication between a professor and the students in a lecture hall. The communication in the second case is mostly monolog; the professor talks most of the time unless a student has a question, a situation in which the protocol dictates that she should raise her hand and wait for permission to speak. In this case, the communication is normally very formal and limited to the subject being taught.

Second Scenario

In the second scenario, we assume that Ann is offered a higher-level position in her company, but needs to move to another branch located in a city very far from Maria. The two friends still want to continue their communication and exchange ideas because

they have come up with an innovative project to start a new business when they both retire. They decide to continue their conversation using regular mail through the post office. However, they do not want their ideas to be revealed by other people if the letters are intercepted. They agree on an encryption/decryption technique. The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter. We discuss the encryption/decryption methods in Chapter 31, but for the moment we assume that Maria and Ann use one technique that makes it hard to decrypt the letter if one does not have the key for doing so. Now we can say that the communication between Maria and Ann takes place in three layers, as shown in Figure 2.2. We assume that Ann and Maria each have three machines (or robots) that can perform the task at each layer.

Figure 2.2 A three-layer protocol



Let us assume that Maria sends the first letter to Ann. Maria talks to the machine at the third layer as though the machine is Ann and is listening to her. The third layer machine listens to what Maria says and creates the plaintext (a letter in English), which is passed to the second layer machine. The second layer machine takes the plaintext, encrypts it, and creates the ciphertext, which is passed to the first layer machine. The first layer machine, presumably a robot, takes the ciphertext, puts it in an envelope, adds the sender and receiver addresses, and mails it.

At Ann's side, the first layer machine picks up the letter from Ann's mail box, recognizing the letter from Maria by the sender address. The machine takes out the ciphertext from the envelope and delivers it to the second layer machine. The second layer machine decrypts the message, creates the plaintext, and passes the plaintext to the third-layer machine. The third layer machine takes the plaintext and reads it as though Maria is speaking.

Protocol layering enables us to divide a complex task into several smaller and simpler tasks. For example, in Figure 2.2, we could have used only one machine to do the job of all three machines. However, if Maria and Ann decide that the encryption/decryption done by the machine is not enough to protect their secrecy, they would have to change the whole machine. In the present situation, they need to change only the second layer machine; the other two can remain the same. This is referred to as *modularity*. Modularity in this case means independent layers. A layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs. If two machines provide the same outputs when given the same inputs, they can replace each other. For example, Ann and Maria can buy the second layer machine from two different manufacturers. As long as the two machines create the same ciphertext from the same plaintext and vice versa, they do the job.

One of the advantages of protocol layering is that it allows us to separate the services from the implementation. A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer; we don't care about how the layer is implemented. For example, Maria may decide not to buy the machine (robot) for the first layer; she can do the job herself. As long as Maria can do the tasks provided by the first layer, in both directions, the communication system works.

Another advantage of protocol layering, which cannot be seen in our simple examples but reveals itself when we discuss protocol layering in the Internet, is that communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

Is there any disadvantage to protocol layering? One can argue that having a single layer makes the job easier. There is no need for each layer to provide a service to the upper layer and give service to the lower layer. For example, Ann and Maria could find or build one machine that could do all three tasks. However, as mentioned above, if one day they found that their code was broken, each would have to replace the whole machine with a new one instead of just changing the machine in the second layer.

2.1.2 Principles of Protocol Layering

Let us discuss two principles of protocol layering.

First Principle

The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction. For example, the third layer task is to listen (in one direction) and *talk* (in the other direction). The second layer needs to be able to encrypt and decrypt. The first layer needs to send and receive mail.

Second Principle

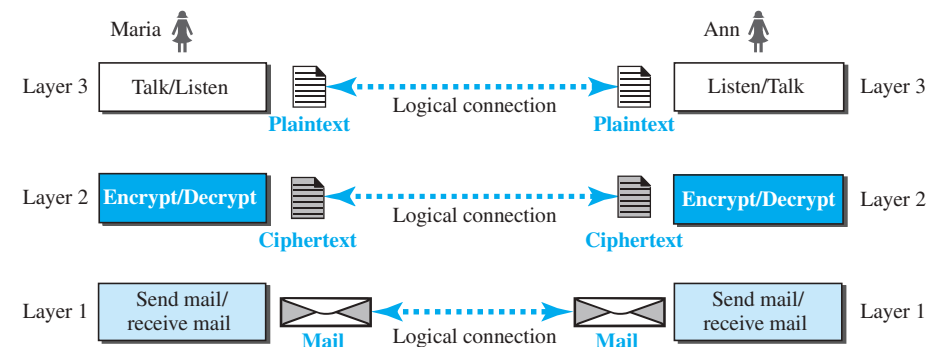
The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical. For example, the object under layer 3 at both sites should be a plaintext letter. The object under layer 2 at

both sites should be a ciphertext letter. The object under layer 1 at both sites should be a piece of mail.

2.1.3 Logical Connections

After following the above two principles, we can think about logical connection between each layer as shown in Figure 2.3. This means that we have layer-to-layer communication. Maria and Ann can think that there is a logical (imaginary) connection at each layer through which they can send the object created from that layer. We will see that the concept of logical connection will help us better understand the task of layering we encounter in data communication and networking.

Figure 2.3 Logical connection between peer layers

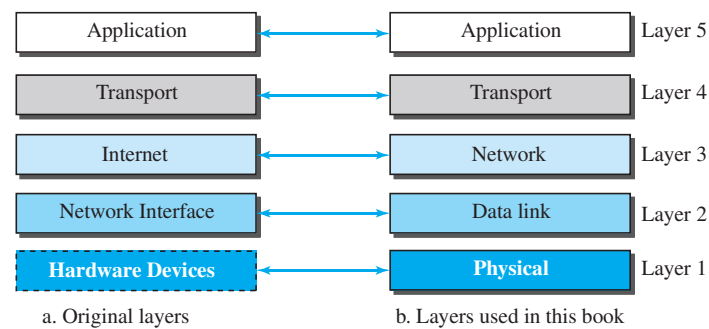
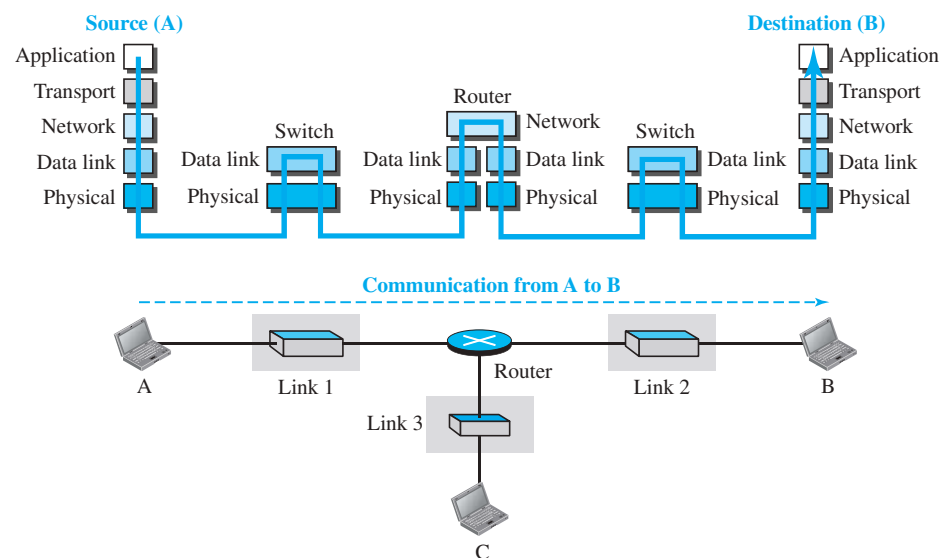


2.2 TCP/IP PROTOCOL SUITE

Now that we know about the concept of protocol layering and the logical communication between layers in our second scenario, we can introduce the TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP is a protocol suite (a set of protocols organized in different layers) used in the Internet today. It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality. The term *hierarchical* means that each upper level protocol is supported by the services provided by one or more lower level protocols. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model. Figure 2.4 shows both configurations.

2.2.1 Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in Figure 2.5.

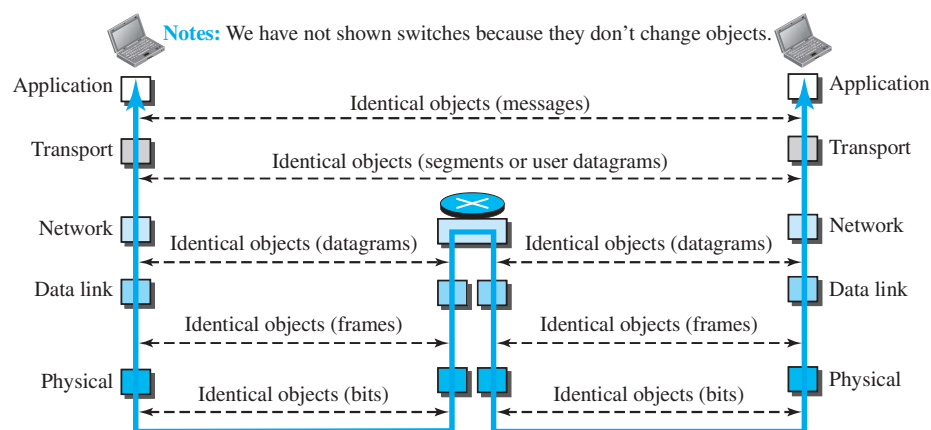
Figure 2.4 *Layers in the TCP/IP protocol suite***Figure 2.5** *Communication through an internet*

Let us assume that computer A communicates with computer B. As the figure shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet. The two hosts are involved in all five layers; the source host needs to create a message in the application layer and send it down the layers so that it is physically sent to the destination host. The destination host needs to receive the communication at the physical layer and then deliver it through the other layers to the application layer.

changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches.

Figure 2.7 shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.

Figure 2.7 *Identical objects in the TCP/IP protocol suite*



Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received (see fragmentation in Chapter 19). Note that the link between two hops does not change the object.

2.2.3 Description of Each Layer

After understanding the concept of logical communication, we are ready to briefly discuss the duty of each layer. Our discussion in this chapter will be very brief, but we come back to the duty of each layer in next five parts of the book.

Physical Layer

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer, the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a *bit*. There are several protocols that transform a bit to a signal. We discuss them in Part II when we discuss the physical layer and the transmission media.

Data-link Layer

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be several overlapping sets of links that a datagram can travel from the host to the destination. The routers are responsible for choosing the *best* links. However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link.

TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called a *frame*.

Each link-layer protocol may provide a different service. Some link-layer protocols provide complete error detection and correction, some provide only error correction. We discuss wired links in Chapters 13 and 14 and wireless links in Chapters 15 and 16.

Network Layer

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes. Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer. One reason, as we said before, is the separation of different tasks between different layers. The second reason is that the routers do not need the application and transport layers. Separating the tasks allows us to use fewer protocols on the routers.

The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer. IP also defines the format and the structure of addresses used in this layer. IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.

IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services. This means that if any of these services is required for an application, the application should rely only on the transport-layer protocol. The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing (it is the responsibility of IP), but it creates forwarding tables for routers to help them in the routing process.

The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks. The Internet Control Message Protocol (ICMP) helps IP to report some problems when routing a packet. The Internet Group Management Protocol (IGMP) is another protocol that helps IP in multitasking. The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host. The Address Resolution Protocol (ARP) is a protocol that helps IP to find the link-layer address of a host or

a router when its network-layer address is given. ARP is discussed in Chapter 9, ICMP in Chapter 19, and IGMP in Chapter 21.

Transport Layer

The logical connection at the transport layer is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport-layer packet (called a *segment* or a *user datagram* in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host. We may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier. The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement.

As we said, there are a few transport-layer protocols in the Internet, each designed for some specific task. The main protocol, Transmission Control Protocol (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data. It creates a logical pipe between two TCPs for transferring a stream of bytes. TCP provides flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network. The other common protocol, User Datagram Protocol (UDP), is a connectionless protocol that transmits user datagrams without first creating a logical connection. In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term *connectionless*). UDP is a simple protocol that does not provide flow, error, or congestion control. Its simplicity, which means small overhead, is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost. A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia. We will discuss UDP, TCP, and SCTP in Chapter 24.

Application Layer

As Figure 2.6 shows, the logical connection between the two application layers is end-to-end. The two application layers exchange *messages* between each other as though there were a bridge between the two layers. However, we should know that the communication is done through all the layers.

Communication at the application layer is between two *processes* (two programs running at this layer). To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer. The application layer in the Internet includes many predefined protocols, but

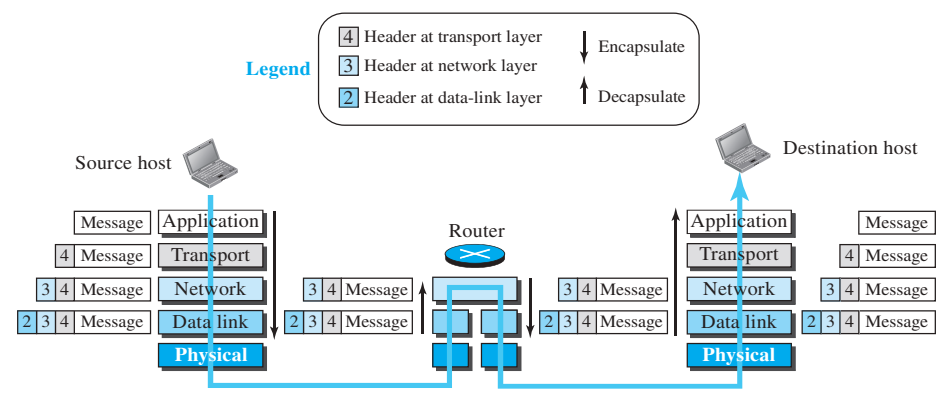
a user can also create a pair of processes to be run at the two hosts. In Chapter 25, we explore this situation.

The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW). The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service. The File Transfer Protocol (FTP) is used for transferring files from one host to another. The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer. The Internet Group Management Protocol (IGMP) is used to collect membership in a group. We discuss most of these protocols in Chapter 26 and some in other chapters.

2.2.4 Encapsulation and Decapsulation

One of the important concepts in protocol layering in the Internet is encapsulation/decapsulation. Figure 2.8 shows this concept for the small internet in Figure 2.5.

Figure 2.8 Encapsulation/Decapsulation



We have not shown the layers for the link-layer switches because no encapsulation/decapsulation occurs in this device. In Figure 2.8, we show the encapsulation in the source host, decapsulation in the destination host, and encapsulation and decapsulation in the router.

Encapsulation at the Source Host

At the source, we have only encapsulation.

1. At the application layer, the data to be exchanged is referred to as a *message*. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the message. The message is passed to the transport layer.
2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that

want to communicate plus some more information that is needed for the end-to-end delivery of the message, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the *segment* (in TCP) and the *user datagram* (in UDP). The transport layer then passes the packet to the network layer.

3. The network layer takes the transport-layer packet as data or payload and adds its own header to the payload. The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on. The result is the network-layer packet, called a *datagram*. The network layer then passes the packet to the data-link layer.
4. The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a *frame*. The frame is passed to the physical layer for transmission.

Decapsulation and Encapsulation at the Router

At the router, we have both decapsulation and encapsulation because the router is connected to two or more links.

1. After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.
2. The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The contents of the datagram should not be changed by the network layer in the router unless there is a need to fragment the datagram if it is too big to be passed through the next link. The datagram is then passed to the data-link layer of the next link.
3. The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

Decapsulation at the Destination Host

At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer. It is necessary to say that decapsulation in the host involves error checking.

2.2.5 Addressing

It is worth mentioning another concept related to protocol layering in the Internet, *addressing*. As we discussed before, we have logical communication between pairs of layers in this model. Any communication that involves two parties needs two addresses: source address and destination address. Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address. Figure 2.9 shows the addressing at each layer.

As the figure shows, there is a relationship between the layer, the address used in that layer, and the packet name at that layer. At the application layer, we normally use names to define the site that provides services, such as *someorg.com*, or the e-mail

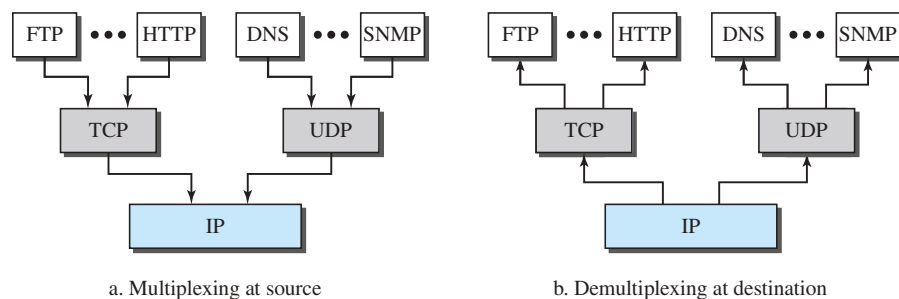
Figure 2.9 Addressing in the TCP/IP protocol suite

Packet names	Layers	Addresses
Message	Application layer	Names
Segment / User datagram	Transport layer	Port numbers
Datagram	Network layer	Logical addresses
Frame	Data-link layer	Link-layer addresses
Bits	Physical layer	

address, such as *somebody@coldmail.com*. At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination. Port numbers are local addresses that distinguish between several programs running at the same time. At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet. The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN). We will come back to these addresses in future chapters.

2.2.6 Multiplexing and Demultiplexing

Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination. Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time); demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time). Figure 2.10 shows the concept of multiplexing and demultiplexing at the three upper layers.

Figure 2.10 Multiplexing and demultiplexing

To be able to multiplex and demultiplex, a protocol needs to have a field in its header to identify to which protocol the encapsulated packets belong. At the transport

layer, either UDP or TCP can accept a message from several application-layer protocols. At the network layer, IP can accept a segment from TCP or a user datagram from UDP. IP can also accept a packet from other protocols such as ICMP, IGMP, and so on. At the data-link layer, a frame may carry the payload coming from IP or other protocols such as ARP (see Chapter 9).

2.3 THE OSI MODEL

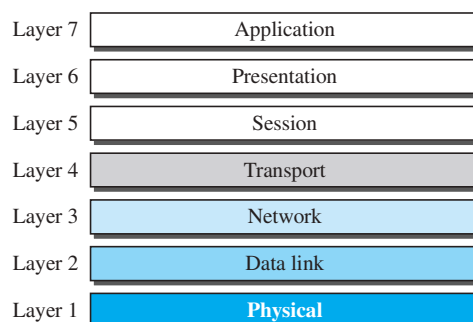
Although, when speaking of the Internet, everyone talks about the TCP/IP protocol suite, this suite is not the only suite of protocols defined. Established in 1947, the **International Organization for Standardization (ISO)** is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of the countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the **Open Systems Interconnection (OSI) model**. It was first introduced in the late 1970s.

ISO is the organization; OSI is the model.

An *open system* is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI model was intended to be the basis for the creation of the protocols in the OSI stack.

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network (see Figure 2.11).

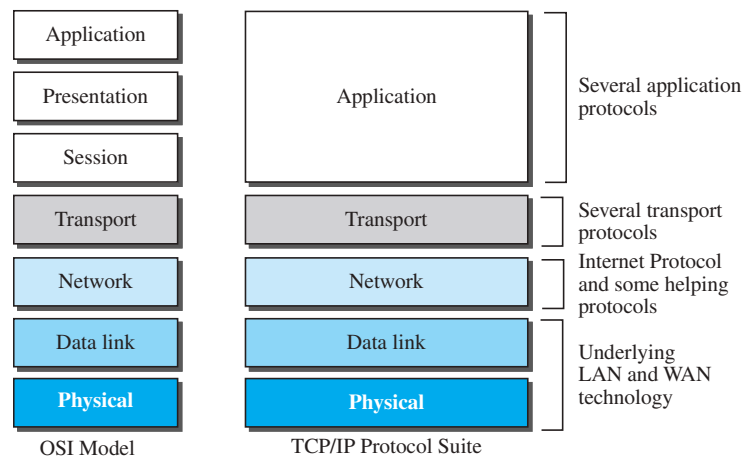
Figure 2.11 *The OSI model*



2.3.1 OSI versus TCP/IP

When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model, as shown in Figure 2.12.

Figure 2.12 TCP/IP and OSI model



Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport-layer protocols. Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation layers are needed for a particular application, they can be included in the development of that piece of software.

2.3.2 Lack of OSI Model's Success

The OSI model appeared after the TCP/IP protocol suite. Most experts were at first excited and thought that the TCP/IP protocol would be fully replaced by the OSI model. This did not happen for several reasons, but we describe only three, which are agreed upon by all experts in the field. First, OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot. Second, some layers in the OSI model were never fully defined. For example, although the services provided by the presentation and the session layers were listed in the document, actual protocols for these two layers were not fully defined, nor were they fully described, and the corresponding software was not fully