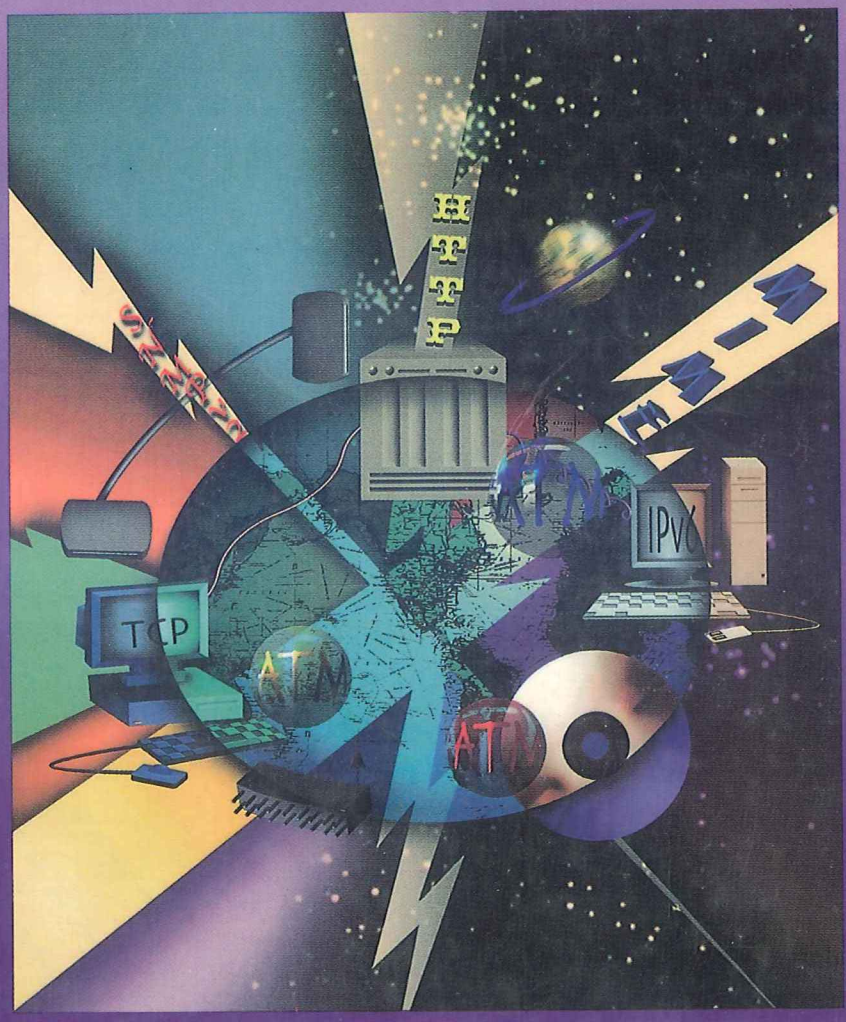
EXHIBIT 1011 – part 1

Eastern Economy Edition

FIFTH EDITION

DATA AND COMPUTER COMMUNICATIONS



WILLIAM STALLINGS

DATA AND COMPUTER COMMUNICATIONS

FIFTH EDITION

DATA AND COMPUTER COMMUNICATIONS

WILLIAM STALLINGS

Prentice-Hall of India Private Limited
New Delhi - 110 001
2001

This Fourteenth Indian Reprint—Rs. 250.00
(Original U.S. Edition—Rs. 3367.00)

DATA AND COMPUTER COMMUNICATIONS, 5th Ed.
by William Stallings

ISBN-81-203-1240-6

Fourteenth Printing (Fifth Edition) ... ... **April, 2001**

Published by Asoke K. Ghosh, Prentice-Hall of India Private Limited, M-97, Connaught Circus, New Delhi-110001 and Printed by Mohan Makhijani at Rekha Printers Private Limited, New Delhi-110020.

As always, for Antigone
and also for her constant
companion, Geoffroi, Chartreux nonpareil

PREFACE

Objectives

This book attempts to provide a unified overview of the broad field of data and computer communications. The organization of the book reflects an attempt to break this massive subject into comprehensible parts and to build, piece by piece, a survey of the state of the art. The book emphasizes basic principles and topics of fundamental importance concerning the technology and architecture of this field, as well as providing a detailed discussion of leading-edge topics.

The following basic themes serve to unify the discussion:

- *Principles:* Although the scope of this book is broad, there are a number of basic principles that appear repeatedly as themes and that unify this field. Examples are multiplexing, flow control, and error control. The book highlights these principles and contrasts their application in specific areas of technology.
- *Design Approaches:* The book examines alternative approaches to meeting specific communication requirements. The discussion is bolstered with examples from existing implementations.
- *Standards:* Standards have come to assume an increasingly important, indeed dominant, role in this field. An understanding of the current status and future direction of technology requires a comprehensive discussion of the role and nature of the related standards.

Plan of the Text

The book is divided into four parts:

I *Data Communications:* This part is concerned primarily with the exchange of data between two directly-connected devices. Within this restricted scope, the key aspects of transmission, interfacing, link control, and multiplexing are examined.

II *Wide-Area Networks:* This part examines the internal mechanisms and technologies that have been developed to support voice, data, and multimedia communications over long-distance networks. The traditional technologies of packet switching and circuit switching are examined, as well as the more recent frame relay and ATM.

III *Local Area Networks:* This part explores the quite different technologies and architectures that have been developed for networking over shorter distances. The transmission media, topologies, and medium access control protocols that are the key ingredients of a LAN design are explored and specific standardized LAN systems examined.

IV *Communications Architecture and Protocols:* This part explores both the architectural principles and the mechanisms required for the exchange of data among computers, workstations, servers, and other data among computers. workstations, servers, and other data processing devices. Much of the material in this part relates to the TCP/IP protocol suite.

In addition, the book includes an extensive glossary, a list of frequently-used acronyms, and a a bibliography. Each chapter includes problems and suggestions for further reading.

The book is intended for both an academic and a professional audience. For the professional interested in this field, the book serves as a basic reference volume and is suitable for self-study.

As a textbook, it can be used for a one-semester or two-semester course. It covers the material in the Computer Communication Networks course of the joint ACM/IEEE Computing Curricula 1991. The chapters and parts of the book are sufficiently modular to provide a great deal of flexibility in the design of courses. The following are suggestions for course design:

- *Fundamentals of Data Communications*: Part I, Chapters 8 (circuit switching), 9 (packet switching), 12 (protocols and architecture).
- *Communications Networks*: If the student has a basic background in data communications, then this course could cover Parts II and III, and Appendix A.
- *Computer Networks*: If the student has a basic background in data communications, then this course could cover Chapters 5 (data communication interface), 6 (data link control), and Part IV.

In addition, a more streamlined course that covers the entire book is possible by eliminating certain chapters that are not essential on a first reading. Chapters that could be optional are: Chapters 2 (data transmission) and 3 (transmission media), if the student has a basic understanding of these topics, Chapter 7 (multiplexing), Chapter 10 (frame relay), Chapter 14 (bridges), and Chapter 18 (network security).

INTERNET SERVICES FOR INSTRUCTORS AND STUDENTS

There is a web page for this book that provides support for students and instructors. The page includes links to relevant sites, transparency masters of figures in the book in PDF (Adobe Acrobat) format, and sign-up information for the book's internet mailing list. The mailing list has been set up so that instructors using this book can exchange information, suggestions, and questions with each other and with the author. The web page is at http://www.shore.net/~ws/DCC5e.html.

As soon as any typos or other errors are discovered, an errata list for this book will be available at http://www.shore.net/~ws/welcome.html.

WHAT'S NEW IN THE FIFTH EDITION

This fifth edition is seeing the light of day less than a dozen years after the publication of the first edition. Much has happened during those years. Indeed, the pace of change, if anything, is increasing. The result is that this revision is more comprehensive and thorough than any of the previous ones. As an indication of this, about one-half of the figures (233 out of 343) and one-half of the tables (48 out of 91) in this edition are new. Every chapter has been revised, new chapters have been added, and the overall organization of the book has changed.

To begin this process of revision, the fourth edition of this book was extensively reviewed by a number of professors who taught from that edition. The result is that, in many places, the narrative has been clarified and tightened and illustrations have been improved. Also, a number of new "field-tested" problems have been added.

Beyond these refinements to improve pedagogy and user-friendliness, there have been major substantive changes throughout the book. Highlights include

- *ATM*: The coverage of ATM has been significantly expanded. There is now an entire chapter devoted to ATM and ATM congestion control (Chapter 11). New to this edition is the coverage of ATM LANs (Sections 13.4 and 14.3).
- *IPv6 (IPng) and IPv6 Security*: IPv6, also known as IPng (next generation), is the key to a greatly expanded use of TCP/IP both on the Internet and in other networks. This new topic is thoroughly covered. The protocol and its internetworking functions are discussed in Section 16.3, and the important material on IPv6 security is provided in Section 18.4.
- *Wireless and Spread Spectrum*: There is greater coverage of wireless technology (Section 3.2) and spread spectrum techniques (Section 4.5). New to this edition is treatment of the important topic of wireless LANs (Sections 12.5 and 13.6).
- *High-speed LANs*: Coverage of this important area is significantly expanded, and includes detailed treatment of leading-edge approaches, including Fast Ethernet (100BASE-T), 100VG-AnyLAN, ATM LANs, and Fibre Channel (Sections 13.1 through 13.5).
- *Routing*: The coverage of internetwork routing has been updated and expanded. There is a longer treatment of OSPF and a discussion of BGP has been added.
- *Frame Relay*: Frame relay also receives expanded coverage with Chapter 10 devoted to frame relay and frame relay congestion control.
- *Network Security*: Coverage of this topic has been expanded to an entire chapter (Chapter 18).
- *Network Management*: New developments in the specification of SNMPv2 are covered (Section 19.2).

- *SMTP and MIME*: Multimedia electronic mail combines the basic functionality of the Simple Mail Transfer Protocol with the Multi-purpose Internet Mail Extension.
- *HTTP*: (Hypertext Transfer Protocol): HTTP is the foundation of the operation of the worldwide web (www). Section 19.3 covers HTTP.
- *TCP/IP*: TCP/IP is now the focus of the protocol coverage in this book. Throughout the book, especially in Part IV, there is increased discussion of TCP/IP and related protocols and issues.

In addition, throughout the book, virtually every topic has been updated to reflect the developments in standards and technology that have occurred since the publication of the second edition.

ACKNOWLEDGMENTS

This new edition has benefited from review by a number of people, who gave generously of their time and expertise. Kitel Albertson (Trondheim College of Engineering), Howard Blum (Pace University), Mike Borella (DePaul University), William Clark (University of Alaska, Anchorage), Joe Doupnik (Utah State University), Doug Jacobson (Iowa State University), Dave Mallya, Biswath Mukherjee (University of California, Davis), and Mark Pullen (George Mason University) reviewed all or part of the manuscript.

Steve Deering of Xerox PARC reviewed the material on IPv6. Ted Doty of Network Systems Corporation reviewed IP security. Henrik Nielson reviewed HTTP.

William Stallings

BRIEF CONTENTS

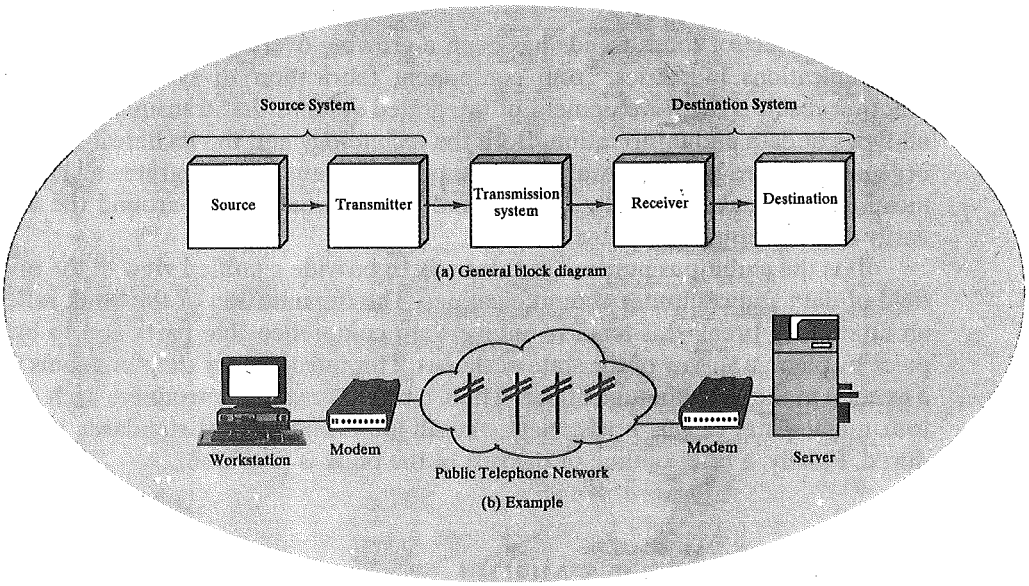
CONTENTS

PART TWO
Wide-Area Networks 229

PART FOUR
Communications Architecture and Protocols 497

CHAPTER 1

INTRODUCTION



The 1970s and 1980s saw a merger of the fields of computer science and data communications that profoundly changed the technology, products, and companies of the now-combined computer-communications industry. Although the consequences of this revolutionary merger are still being worked out, it is safe to say that the revolution has occurred, and any investigation of the field of data communications must be made within this new context.

The computer-communications revolution has produced several remarkable facts:

- There is no fundamental difference between data processing (computers) and data communications (transmission and switching equipment).
- There are no fundamental differences among data, voice, and video communications.
- The lines between single-processor computer, multi-processor computer, local network, metropolitan network, and long-haul network have blurred.

One effect of these trends has been a growing overlap of the computer and communications industries, from component fabrication to system integration. Another result is the development of integrated systems that transmit and process all types of data and information. Both the technology and the technical-standards organizations are driving toward a single public system that integrates all communications and makes virtually all data and information sources around the world easily and uniformly accessible.

It is the ambitious purpose of this book to provide a unified view of the broad field of data and computer communications. The organization of the book reflects an attempt to break this massive subject into comprehensible parts and to build, piece by piece, a survey of the state of the art. This introductory chapter begins with a general model of communications. Then, a brief discussion introduces each of the four major parts of this book. Next, the all-important role of standards is introduced. Finally, a brief outline of the rest of the book is provided.

1.1 A COMMUNICATIONS MODEL

We begin our study with a simple model of communications, illustrated by the block diagram in Figure 1.1a.

The fundamental purpose of a communications system is the exchange of data between two parties. Figure 1.1b presents one particular example, which is the communication between a workstation and a server over a public telephone network. Another example is the exchange of voice signals between two telephones over the same network. The key elements of the model are

Source. This device generates the data to be transmitted; examples are telephones and personal computers.

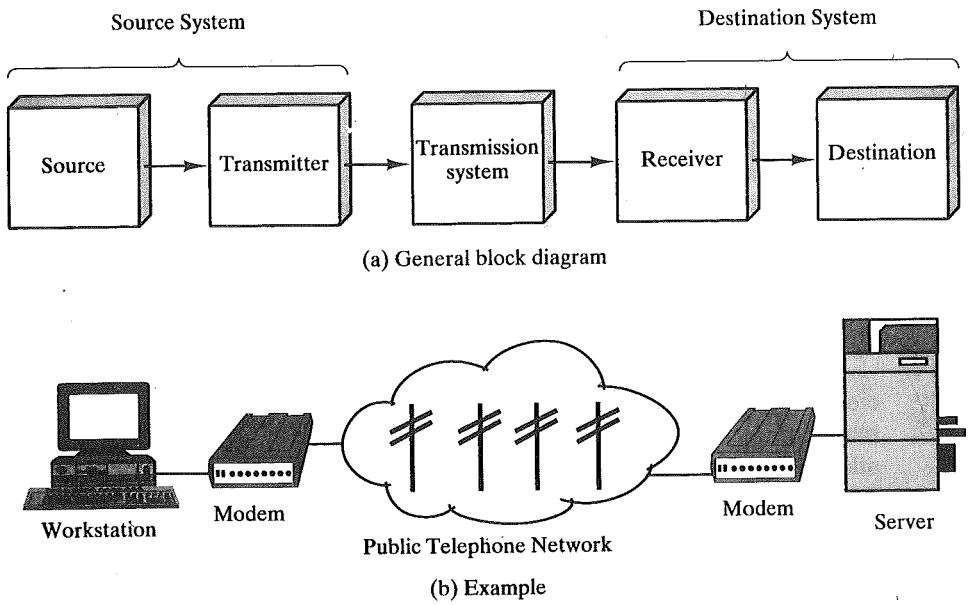


FIGURE 1.1 Simplified communications model.

- **Transmitter.** Usually, the data generated by a source system are not transmitted directly in the form in which they were generated. Rather, a transmitter transforms and encodes the information in such a way as to produce electromagnetic signals that can be transmitted across some sort of transmission system. For example, a modem takes a digital bit stream from an attached device such as a personal computer and transforms that bit stream into an analog signal that can be handled by the telephone network.
- **Transmission System.** This can be a single transmission line or a complex network connecting source and destination.
- **Receiver.** The receiver accepts the signal from the transmission system and converts it into a form that can be handled by the destination device. For example, a modem will accept an analog signal coming from a network or transmission line and convert it into a digital bit stream.
- **Destination.** Takes the incoming data from the receiver.

This simple narrative conceals a wealth of technical complexity. To get some idea of the scope of this complexity, Table 1.1 lists some of the key tasks that must be performed in a data communications system. The list is somewhat arbitrary: Elements could be added; items on the list could be merged; and some items represent several tasks that are performed at different "levels" of the system. However, the list as it stands is suggestive of the scope of this book.

TABLE 1.1 Communications tasks.

Transmission system utilization	Addressing
Interfacing	Routing
Signal generation	Recovery
Synchronization	Message formatting
Exchange management	Security
Error detection and correction	Network management
Flow control	

The first item, **transmission system utilization**, refers to the need to make efficient use of transmission facilities that are typically shared among a number of communicating devices. Various techniques (referred to as multiplexing) are used to allocate the total capacity of a transmission medium among a number of users. Congestion control techniques may be required to assure that the system is not overwhelmed by excessive demand for transmission services.

In order to communicate, a device must **interface** with the transmission system. All the forms of communication discussed in this book depend, at bottom, on the use of electromagnetic signals propagated over a transmission medium. Thus, once an interface is established, **signal generation** is required for communication. The properties of the signal, such as form and intensity, must be such that they are (1) capable of being propagated through the transmission system, and (2) interpretable as data at the receiver.

Not only must the signals be generated to conform to the requirements of the transmission system and receiver, but there must be some form of **synchronization** between transmitter and receiver. The receiver must be able to determine when a signal begins to arrive and when it ends. It must also know the duration of each signal element.

Beyond the basic matter of deciding on the nature and timing of signals, there are a variety of requirements for communication between two parties that might be collected under the term **exchange management**. If data are to be exchanged in both directions over a period of time, the two parties must cooperate. For example, for two parties to engage in a telephone conversation, one party must dial the number of the other, causing signals to be generated that result in the ringing of the called phone. The called party completes a connection by lifting the receiver. For data processing devices, more will be needed than simply establishing a connection; certain conventions must be decided upon. These conventions may include whether both devices may transmit simultaneously or must take turns, the amount of data to be sent at one time, the format of the data, and what to do if certain contingencies, such as an error, arise.

The next two items might have been included under exchange management, but they are important enough to list separately. In all communications systems, there is a potential for error; transmitted signals are distorted to some extent before reaching their destination. **Error detection and correction** are required in circumstances where errors cannot be tolerated; this is usually the case with data process-

ing systems. For example, in transferring a file from one computer to another, it is simply not acceptable for the contents of the file to be accidentally altered. **Flow control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed.

Next, we mention the related but distinct concepts of **addressing** and **routing**. When a transmission facility is shared by more than two devices, a source system must somehow indicate the identity of the intended destination. The transmission system must assure that the destination system, and only that system, receives the data. Further, the transmission system may itself be a network through which various paths may be taken. A specific route through this network must be chosen.

Recovery is a concept distinct from that of error correction. Recovery techniques are needed in situations in which an information exchange, such as a data base transaction or file transfer, is interrupted due to a fault somewhere in the system. The objective is either to be able to resume activity at the point of interruption or at least to restore the state of the systems involved to the condition prior to the beginning of the exchange.

Message formatting has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted. For example, both sides must use the same binary code for characters.

Frequently, it is important to provide some measure of **security** in a data communications system. The sender of data may wish to be assured that only the intended party actually receives the data; and the receiver of data may wish to be assured that the received data have not been altered in transit and that the data have actually come from the purported sender.

Finally, a data communications facility is a complex system that cannot create or run itself. **Network management** capabilities are needed to configure the system, monitor its status, react to failures and overloads, and plan intelligently for future growth.

Thus we have gone from the simple idea of data communication between source and destination to a rather formidable list of data communications tasks. In this book, we further elaborate this list of tasks to describe and encompass the entire set of activities that can be classified under data and computer communications.

1.2 DATA COMMUNICATIONS

This book is organized into four parts. The first part deals with the most fundamental aspects of the communications function, focusing on the transmission of signals in a reliable and efficient manner. For want of a better name, we have given Part I the title "Data Communications," although that term arguably encompasses some or even all of the topics of Parts II, III, and IV.

To get some flavor for the focus of Part I, Figure 1.2 provides a new perspective on the communications model of Figure 1.1a. Let us trace through the details of this figure using electronic mail as an example.

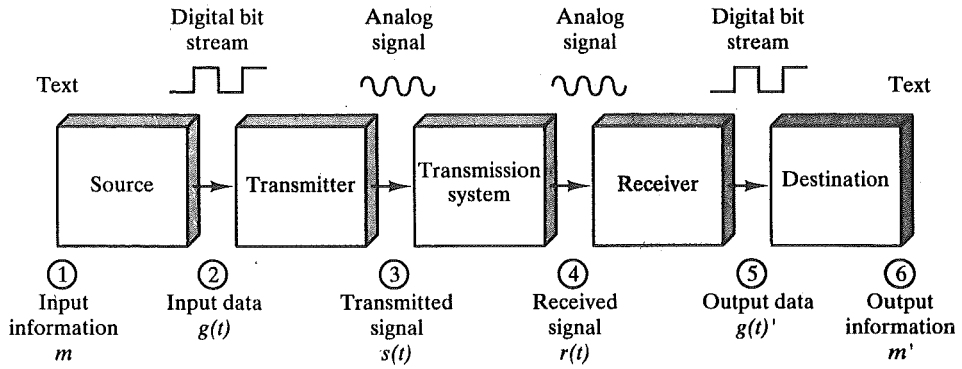


FIGURE 1.2 Simplified data communications model.

Consider that the input device and transmitter are components of a personal computer. The user of the PC wishes to send a message to another user—for example, "The meeting scheduled for March 25 is canceled" (m). The user activates the electronic mail package on the PC and enters the message via the keyboard (input device). The character string is briefly buffered in main memory. We can view it as a sequence of bits (g) in memory. The personal computer is connected to some transmission medium, such as a local network or a telephone line, by an I/O device (transmitter), such as a local network transceiver or a modem. The input data are transferred to the transmitter as a sequence of voltage shifts [$g(t)$] representing bits on some communications bus or cable. The transmitter is connected directly to the medium and converts the incoming stream [$g(t)$] into a signal [$s(t)$] suitable for transmission. Specific alternatives to this procedure will be described in Chapter 4.

The transmitted signal $s(t)$ presented to the medium is subject to a number of impairments, discussed in Chapter 2, before it reaches the receiver. Thus, the received signal $r(t)$ may differ to some degree from $s(t)$. The receiver will attempt to estimate the original $s(t)$, based on $r(t)$ and its knowledge of the medium, producing a sequence of bits $g'(t)$. These bits are sent to the output personal computer, where they are briefly buffered in memory as a block of bits (g). In many cases, the destination system will attempt to determine if an error has occurred and, if so, will cooperate with the source system to eventually obtain a complete, error-free block of data. These data are then presented to the user via an output device, such as a printer or a screen. The message (m'), as viewed by the user, will usually be an exact copy of the original message (m).

Now consider a telephone conversation. In this case, the input to the telephone is a message (m) in the form of sound waves. The sound waves are converted by the telephone into electrical signals of the same frequency. These signals are transmitted without modification over the telephone line. Hence, the input signal $g(t)$ and the transmitted signal $s(t)$ are identical. The signal $s(t)$ will suffer some distortion over the medium, so that $r(t)$ will not be identical to $s(t)$. Nevertheless, the signal $r(t)$ is converted back into a sound wave with no attempt at correction or

improvement of signal quality. Thus m is not an exact replica of m. However, the received sound message is generally comprehensible to the listener.

The discussion so far does not touch on other key aspects of data communications, including data-link control techniques for controlling the flow of data and detecting and correcting errors, and multiplexing techniques for transmission efficiency. All of these topics are explored in Part I.

1.3 DATA COMMUNICATIONS NETWORKING

In its simplest form, data communication takes place between two devices that are directly connected by some form of point-to-point transmission medium. Often, however, it is impractical for two devices to be directly, point-to-point connected. This is so for one (or both) of the following contingencies:

- The devices are very far apart. It would be inordinately expensive, for example, to string a dedicated link between two devices thousands of miles apart.
- There is a set of devices, each of which may require a link to many of the others at various times. Examples are all of the telephones in the world and all of the terminals and computers owned by a single organization. Except for the case of a very few devices, it is impractical to provide a dedicated wire between each pair of devices.

The solution to this problem is to attach each device to a communications network. Figure 1.3 relates this area to the communications model of Figure 1.1a and also suggests the two major categories into which communications networks are traditionally classified: wide-area networks (WANs) and local-area networks (LANs). The distinction between the two, both in terms of technology and application, has become somewhat blurred in recent years, but it remains a useful way of organizing the discussion.

Wide-Area Networks

Wide-area networks have been traditionally been considered to be those that cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes (including the boundary nodes) are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination.

Traditionally, WANs have been implemented using one of two technologies: circuit switching and packet switching. More recently, frame relay and ATM networks have assumed major roles.

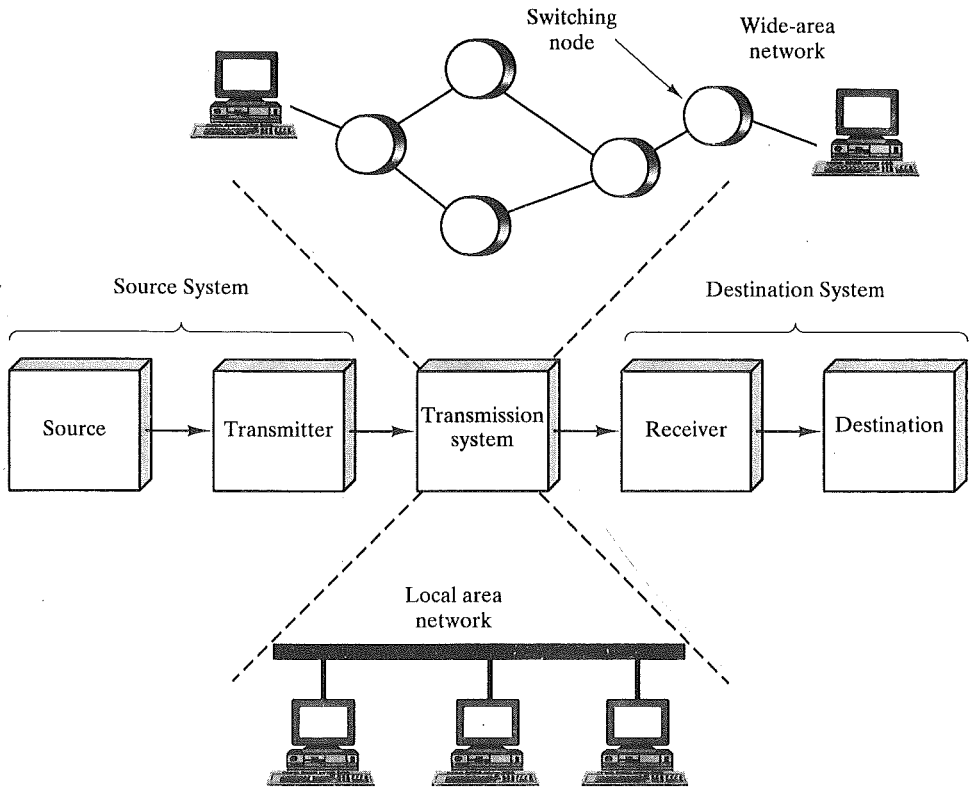


FIGURE 1.3 Simplified network models.

Circuit Switching

In a circuit-switched network, a dedicated communications path is established between two stations through the nodes of the network. That path is a connected sequence of physical links between nodes. On each link, a logical channel is dedicated to the connection. Data generated by the source station are transmitted along the dedicated path as rapidly as possible. At each node, incoming data are routed or switched to the appropriate outgoing channel without delay. The most common example of circuit switching is the telephone network.

Packet Switching

A quite different approach is used in a packet-switched network. In this case, it is not necessary to dedicate transmission capacity along a path through the network. Rather, data are sent out in a sequence of small chunks, called packets. Each packet is passed through the network from node to node along some path leading from source to destination. At each node, the entire packet is received, stored briefly, and then transmitted to the next node. Packet-switched networks are commonly used for terminal-to-computer and computer-to-computer communications.

Frame Relay

Packet switching was developed at a time when digital long-distance transmission facilities exhibited a relatively high error rate compared to today's facilities. As a result, there is a considerable amount of overhead built into packet-switched schemes to compensate for errors. The overhead includes additional bits added to each packet to introduce redundancy and additional processing at the end stations and the intermediate switching nodes to detect and recover from errors.

With modern high-speed telecommunications systems, this overhead is unnecessary and counterproductive. It is unnecessary because the rate of errors has been dramatically lowered and any remaining errors can easily be caught in the end systems by logic that operates above the level of the packet-switching logic; it is counterproductive because the overhead involved soaks up a significant fraction of the high capacity provided by the network.

Frame relay was developed to take advantage of these high data rates and low error rates. Whereas the original packet-switching networks were designed with a data rate to the end user of about 64 kbps, frame relay networks are designed to operate efficiently at user data rates of up to 2 Mbps. The key to achieving these high data rates is to strip out most of the overhead involved with error control.

ATM

Asynchronous transfer mode (ATM), sometimes referred to as cell relay, is a culmination of all of the developments in circuit switching and packet switching over the past 25 years.

ATM can be viewed as an evolution from frame relay. The most obvious difference between frame relay and ATM is that frame relay uses variable-length packets, called frames, and ATM uses fixed-length packets, called cells. As with frame relay, ATM provides little overhead for error control, depending on the inherent reliability of the transmission system and on higher layers of logic in the end systems to catch and correct errors. By using a fixed-packet length, the processing overhead is reduced even further for ATM compared to frame relay. The result is that ATM is designed to work in the range of 10s and 100s of Mbps, compared to the 2-Mbps target of frame relay.

ATM can also be viewed as an evolution from circuit switching. With circuit-switching, only fixed-data-rate circuits are available to the end system. ATM allows the definition of multiple virtual channels with data rates that are dynamically defined at the time the virtual channel is created. By using full, fixed-size cells, ATM is so efficient that it can offer a constant-data-rate channel even though it is using a packet-switching technique. Thus, ATM extends circuit switching to allow multiple channels with the data rate on each channel dynamically set on demand.

ISDN and Broadband ISDN

Merging and evolving communications and computing technologies, coupled with increasing demands for efficient and timely collection, processing, and dissemination of information, are leading to the development of integrated systems that

transmit and process all types of data. A significant outgrowth of these trends is the integrated services digital network (ISDN).

The ISDN is intended to be a worldwide public telecommunications network to replace existing public telecommunications networks and deliver a wide variety of services. The ISDN is defined by the standardization of user interfaces and implemented as a set of digital switches and paths supporting a broad range of traffic types and providing value-added processing services. In practice, there are multiple networks, implemented within national boundaries, but, from the user's point of view, there is intended to be a single, uniformly accessible, worldwide network.

Despite the fact that ISDN has yet to achieve the universal deployment hoped for, it is already in its second generation. The first generation, sometimes referred to as **narrowband ISDN**, is based on the use of a 64-kbps channel as the basic unit of switching and has a circuit-switching orientation. The major technical contribution of the narrowband ISDN effort has been frame relay. The second generation, referred to as **broadband ISDN**, supports very high data rates (100s of Mbps) and has a packet-switching orientation. The major technical contribution of the broadband ISDN effort has been asynchronous transfer mode (ATM), also known as cell relay.

Local Area Networks

As with wide-area networks, a local-area network is a communications network that interconnects a variety of devices and provides a means for information exchange among those devices. There are several key distinctions between LANs and WANs:

1. The scope of the LAN is small, typically a single building or a cluster of buildings. This difference in geographic scope leads to different technical solutions, as we shall see.
2. It is usually the case that the LAN is owned by the same organization that owns the attached devices. For WANs, this is less often the case, or at least a significant fraction of the network assets are not owned. This has two implications. First, care must be taken in the choice of LAN, as there may be a substantial capital investment (compared to dial-up or leased charges for wide-area networks) for both purchase and maintenance. Second, the network management responsibility for a local network falls solely on the user.
3. The internal data rates of LANs are typically much greater than those of wide-area networks.

Traditionally, LANs make use of a broadcast network approach rather than a switching approach. With a broadcast communication network, there are no intermediate switching nodes. At each station, there is a transmitter/receiver that communicates over a medium shared by other stations. A transmission from any one station is broadcast to and received by all other stations. A simple example of this is a CB radio system, in which all users tuned to the same channel may communicate. We will be concerned with networks used to link computers, workstations, and

other digital devices. In the latter case, data are usually transmitted in packets. Because the medium is shared, only one station at a time can transmit a packet.

More recently, examples of switched LANs have appeared. The two most prominent examples are ATM LANs, which simply use an ATM network in a local area, and Fibre Channel. We will examine these LANs, as well as the more common broadcast LANs, in Part III.

1.4 PROTOCOLS AND PROTOCOL ARCHITECTURE

When computers, terminals, and/or other data processing devices exchange data, the scope of concern is much broader than the concerns we have discussed in Sections 1.2 and 1.3. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed. Typical tasks to be performed are

1. The source system must either activate the direct data communication path or inform the communication network of the identity of the desired destination system.
2. The source system must ascertain that the destination system is prepared to receive data.
3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user.
4. If the file formats used on the two systems are incompatible, one or the other system must perform a format translation function.

It is clear that there must be a high degree of cooperation between the two computer systems. The exchange of information between computers for the purpose of cooperative action is generally referred to as *computer communications*. Similarly, when two or more computers are interconnected via a communication network, the set of computer stations is referred to as a *computer network*. Because a similar level of cooperation is required between a user at a terminal and one at a computer, these terms are often used when some of the communicating entities are terminals.

In discussing computer communications and computer networks, two concepts are paramount:

- Protocols
- Computer-communications architecture, or protocol architecture

A protocol is used for communication between entities in different systems. The terms "entity" and "system" are used in a very general sense. Examples of

entities are user application programs, file transfer packages, data-base management systems, electronic mail facilities, and terminals. Examples of systems are computers, terminals, and remote sensors. Note that in some cases the entity and the system in which it resides are coextensive (e.g., terminals). In general, an entity is anything capable of sending or receiving information, and a system is a physically distinct object that contains one or more entities. For two entities to communicate successfully, they must "speak the same language." What is communicated, how it is communicated, and when it is communicated must conform to some mutually acceptable conventions between the entities involved. The conventions are referred to as a protocol, which may be defined as a set of rules governing the exchange of data between two entities. The key elements of a protocol are

- **Syntax.** Includes such things as data format and signal levels.
- **Semantics.** Includes control information for coordination and error handling.
- **Timing.** Includes speed matching and sequencing.

Having introduced the concept of a protocol, we can now introduce the concept of a protocol architecture. It is clear that there must be a high degree of cooperation between the two computers. Instead of implementing the logic for this as a single module, the task is broken up into subtasks, each of which is implemented separately. As an example, Figure 1.4 suggests the way in which a file transfer facility could be implemented. Three modules are used. Tasks 3 and 4 in the preceding list could be performed by a file transfer module. The two modules on the two systems exchange files and commands. However, rather than requiring the file transfer module to handle the details of actually transferring data and commands, the file transfer modules each rely on a communications service module. This module is responsible for making sure that the file transfer commands and data are reliably exchanged between systems. Among other things, this module would perform task 2. Now, the nature of the exchange between systems is independent of the nature of the network that interconnects them. Therefore, rather than building details of the network interface into the communications service module, it makes sense to have a third module, a network access module, that performs task 1 by interacting with the network.

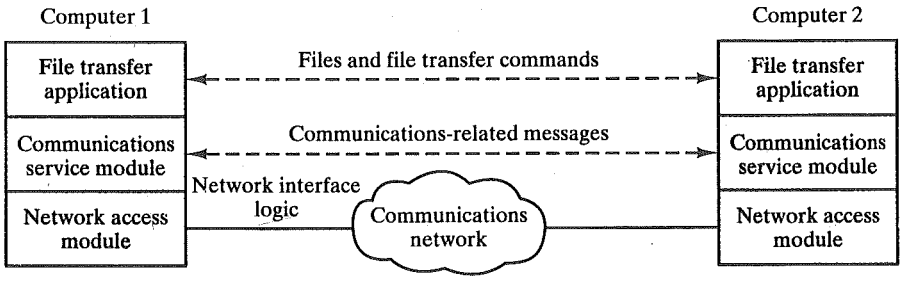


FIGURE 1.4 A simplified architecture for file transfer.

Let us try to summarize the motivation for the three modules in Figure 1.4. The file transfer module contains all of the logic that is unique to the file transfer application, such as transmitting passwords, file commands, and file records. There is a need to transmit these files and commands reliably. However, the same sorts of reliability requirements are relevant to a variety of applications (e.g., electronic mail, document transfer). Therefore, these requirements are met by a separate communications service module that can be used by a variety of applications. The communications service module is concerned with assuring that the two computer systems are active and ready for data transfer and for keeping track of the data that are being exchanged to assure delivery. However, these tasks are independent of the type of network that is being used. Therefore, the logic for actually dealing with the network is separated out into a separate network access module. That way, if the network to be used is changed, only the network access module is affected.

Thus, instead of a single module for performing communications, there is a structured set of modules that implements the communications function. That structure is referred to as a protocol architecture. In the remainder of this section, we generalize the preceding example to present a simplified protocol architecture. Following that, we look at more complex, real-world examples: TCP/IP and OSI.

A Three-Layer Model

In very general terms, communications can be said to involve three agents: applications, computers, and networks. One example of an application is a file transfer operation. These applications execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting it to the intended application within the computer.

With these concepts in mind, it appears natural to organize the communication task into three relatively independent layers:

- Network access layer
- Transport layer
- Application layer

The **network access layer** is concerned with the exchange of data between a computer and the network to which it is attached. The sending computer must provide the network with the address of the destination computer, so that the network may route the data to the appropriate destination. The sending computer may wish to invoke certain services, such as priority, that might be provided by the network. The specific software used at this layer depends on the type of network to be used; different standards have been developed for circuit switching, packet switching, local area networks, and others. Thus, it makes sense to separate those functions having to do with network access into a separate layer. By doing this, the remainder of the communications software, above the network access layer, need not be

concerned with the specifics of the network to be used. The same higher-layer software should function properly regardless of the particular network to which the computer is attached.

Regardless of the nature of the applications that are exchanging data, there is usually a requirement that data be exchanged reliably. That is, we would like to be assured that all of the data arrive at the destination application and that the data arrive in the same order in which they were sent. As we shall see, the mechanisms for providing reliability are essentially independent of the nature of the applications. Thus, it makes sense to collect those mechanisms in a common layer shared by all applications; this is referred to as the **transport layer**.

Finally, the **application layer** contains the logic needed to support the various user applications. For each different type of application, such as file transfer, a separate module is needed that is peculiar to that application.

Figures 1.5 and 1.6 illustrate this simple architecture. Figure 1.5 shows three computers connected to a network. Each computer contains software at the network access and transport layers and software at the application layer for one or more applications. For successful communication, every entity in the overall system must have a unique address. Actually, two levels of addressing are needed. Each computer on the network must have a unique network address; this allows the network to deliver data to the proper computer. Each application on a computer must have an address that is unique within that computer; this allows the transport layer to support multiple applications at each computer. These latter addresses are

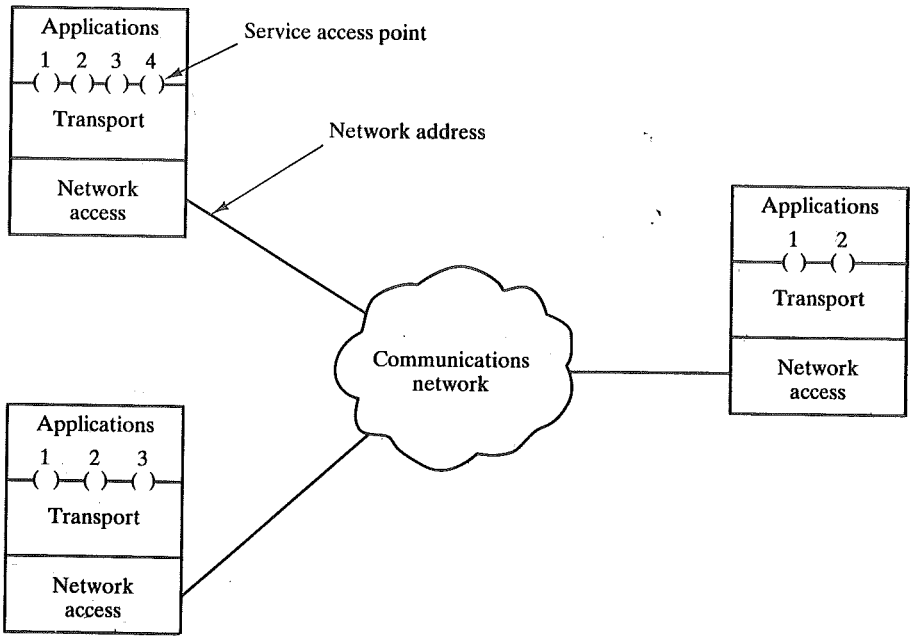


FIGURE 1.5 Protocol architectures and networks.

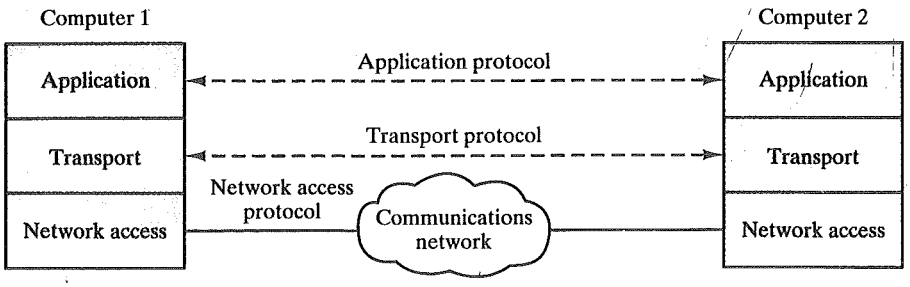


FIGURE 1.6 Protocols in a simplified architecture.

known as service access points (SAPs), connoting that each application is individually accessing the services of the transport layer.

Figure 1.6 indicates the way in which modules at the same level on different computers communicate with each other: by means of a protocol. A protocol is the set of rules or conventions governing the ways in which two entities cooperate to exchange data. A protocol specification details the control functions that may be performed, the formats and control codes used to communicate those functions, and the procedures that the two entities must follow.

Let us trace a simple operation. Suppose that an application, associated with SAP 1 at computer A, wishes to send a message to another application, associated with SAP 2 at computer B. The application at A hands the message over to its transport layer with instructions to send it to SAP 2 on computer B. The transport layer hands the message over to the network access layer, which instructs the network to send the message to computer B. Note that the network need not be told the identity of the destination service access point. All that it needs to know is that the data are intended for computer B.

To control this operation, control information, as well as user data, must be transmitted, as suggested in Figure 1.7. Let us say that the sending application generates a block of data and passes this to the transport layer. The transport layer may break this block into two smaller pieces to make it more manageable. To each of these pieces the transport layer appends a transport header, containing protocol control information. The combination of data from the next higher layer and control information is known as a protocol data unit (PDU); in this case, it is referred to as a transport protocol data unit. The header in each transport PDU contains control information to be used by the peer transport protocol at computer B. Examples of items that may be stored in this header include

- **Destination SAP.** When the destination transport layer receives the transport protocol data unit, it must know to whom the data are to be delivered.
- **Sequence number.** Because the transport protocol is sending a sequence of protocol data units, it numbers them sequentially so that if they arrive out of order, the destination transport entity may reorder them.

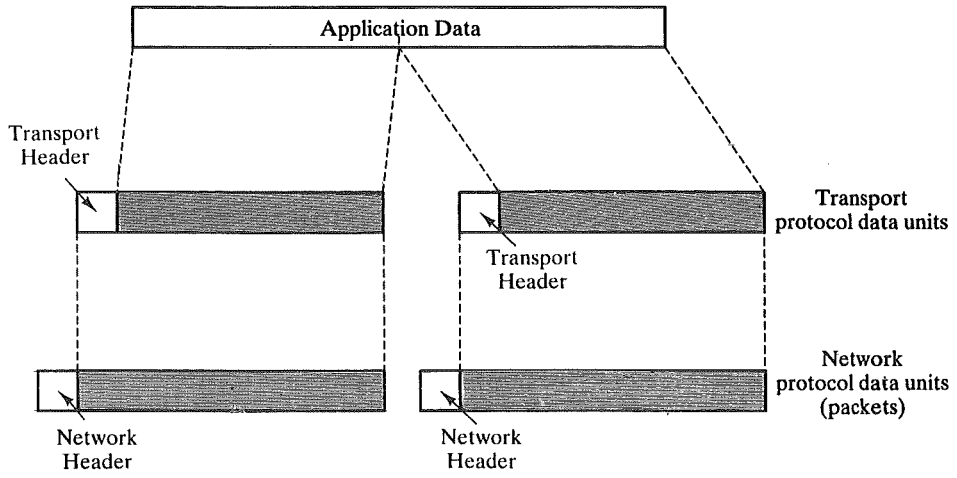


FIGURE 1.7 Protocol data units.

- **Error-detection code.** The sending transport entity may include a code that is a function of the contents of the remainder of the PDU. The receiving transport protocol performs the same calculation and compares the result with the incoming code. A discrepancy results if there has been some error in transmission. In that case, the receiver can discard the PDU and take corrective action.

The next step is for the transport layer to hand each protocol data unit over to the network layer, with instructions to transmit it to the destination computer. To satisfy this request, the network access protocol must present the data to the network with a request for transmission. As before, this operation requires the use of control information. In this case, the network access protocol appends a network access header to the data it receives from the transport layer, creating a network-access PDU. Examples of the items that may be stored in the header include

- **Destination computer address.** The network must know to whom (which computer on the network) the data are to be delivered.
- **Facilities requests.** The network access protocol might want the network to make use of certain facilities, such as priority.

Figure 1.8 puts all of these concepts together, showing the interaction between modules to transfer one block of data. Let us say that the file transfer module in computer X is transferring a file one record at a time to computer Y. Each record is handed over to the transport layer module. We can picture this action as being in the form of a command or procedure call. The arguments of this procedure call include the destination computer address, the destination service access point, and

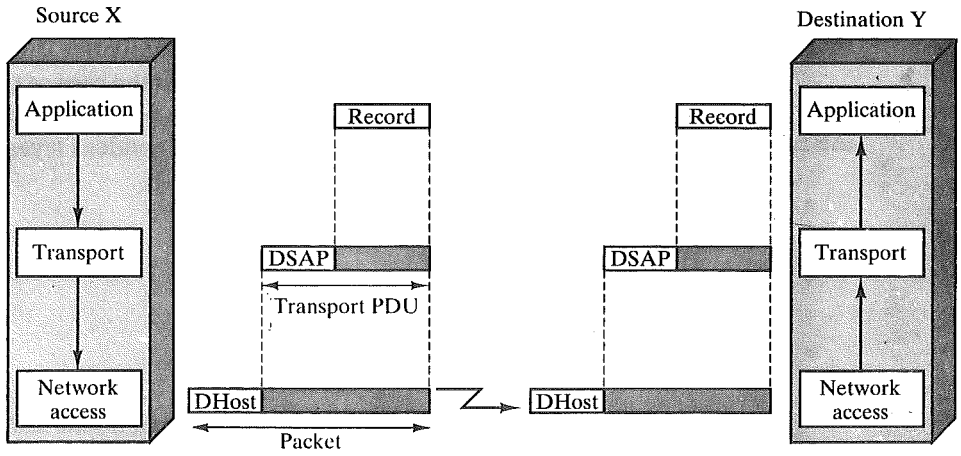


FIGURE 1.8 Operation of a protocol architecture.

the record. The transport layer appends the destination service access point and other control information to the record to create a transport PDU. This is then handed down to the network access layer by another procedure call. In this case, the arguments for the command are the destination computer address and the transport protocol data unit. The network access layer uses this information to construct a network PDU. The transport protocol data unit is the data field of the network PDU, and the network PDU header includes information concerning the source and destination computer addresses. Note that the transport header is not "visible" at the network access layer; the network access layer is not concerned with the contents of the transport PDU.

The network accepts the network PDU from X and delivers it to Y. The network access module in Y receives the PDU, strips off the header, and transfers the enclosed transport PDU to X's transport layer module. The transport layer examines the transport protocol data unit header and, on the basis of the SAP field in the header, delivers the enclosed record to the appropriate application, in this case the file transfer module in Y.

The TCP/IP Protocol Architecture

Two protocol architectures have served as the basis for the development of interoperable communications standards: the TCP/IP protocol suite and the OSI reference model. TCP/IP is the most widely used interoperable architecture, and OSI has become the standard model for classifying communications functions. In the remainder of this section, we provide a brief overview of the two architectures; the topic is explored more fully in Chapter 15.

TCP/IP is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the

TCP/IP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Activities Board (IAB).

There is no official TCP/IP protocol model as there is in the case of OSI. However, based on the protocol standards that have been developed, we can organize the communication task for TCP/IP into five relatively independent layers:

- Application layer
- Host-to-host, or transport layer
- Internet layer
- Network access layer
- Physical layer

The **physical layer** covers the physical interface between a data transmission device (e.g., workstation, computer) and a transmission medium or network. This layer is concerned with specifying the characteristics of the transmission medium, the nature of the signals, the data rate, and related matters.

The **network access layer** is concerned with the exchange of data between an end system and the network to which it is attached. The sending computer must provide the network with the address of the destination computer, so that the network may route the data to the appropriate destination. The sending computer may wish to invoke certain services, such as priority, that might be provided by the network. The specific software used at this layer depends on the type of network to be used; different standards have been developed for circuit-switching, packet-switching (e.g., X.25), local area networks (e.g., Ethernet), and others. Thus, it makes sense to separate those functions having to do with network access into a separate layer. By doing this, the remainder of the communications software, above the network access layer, need not be concerned about the specifics of the network to be used. The same higher-layer software should function properly regardless of the particular network to which the computer is attached.

The network access layer is concerned with access to and routing data across a network for two end systems attached to the same network. In those cases where two devices are attached to different networks, procedures are needed to allow data to traverse multiple interconnected networks. This is the function of the **internet layer**. The internet protocol (IP) is used at this layer to provide the routing function across multiple networks. This protocol is implemented not only in the end systems but also in routers. A router is a processor that connects two networks and whose primary function is to relay data from one network to the other on its route from the source to the destination end system.

Regardless of the nature of the applications that are exchanging data, there is usually a requirement that data be exchanged reliably. That is, we would like to be assured that all of the data arrive at the destination application and that the data arrive in the same order in which they were sent. As we shall see, the mechanisms for providing reliability are essentially independent of the nature of the applications. Thus, it makes sense to collect those mechanisms in a common layer shared

by all applications; this is referred to as the **host-to-host layer**, or **transport layer**. The transmission control protocol (TCP) is the most commonly-used protocol to provide this functionality.

Finally, the **application layer** contains the logic needed to support the various user applications. For each different type of application, such as file transfer, a separate module is needed that is peculiar to that application.

Figure 1.9 shows how the TCP/IP protocols are implemented in end systems and relates this description to the communications model of Figure 1.1a. Note that the physical and network access layers provide interaction between the end system and the network, whereas the transport and application layers are what is known as end-to-end protocols; they support interaction between two end systems. The internet layer has the flavor of both. At this layer, the end system communicates routing information to the network but also must provide some common functions between the two end systems; these will be explored in Chapters 15 and 16.

The OSI Model

The open systems interconnection (OSI) model was developed by the International Organization for Standardization (ISO) as a model for a computer communications architecture and as a framework for developing protocol standards. It consists of seven layers:

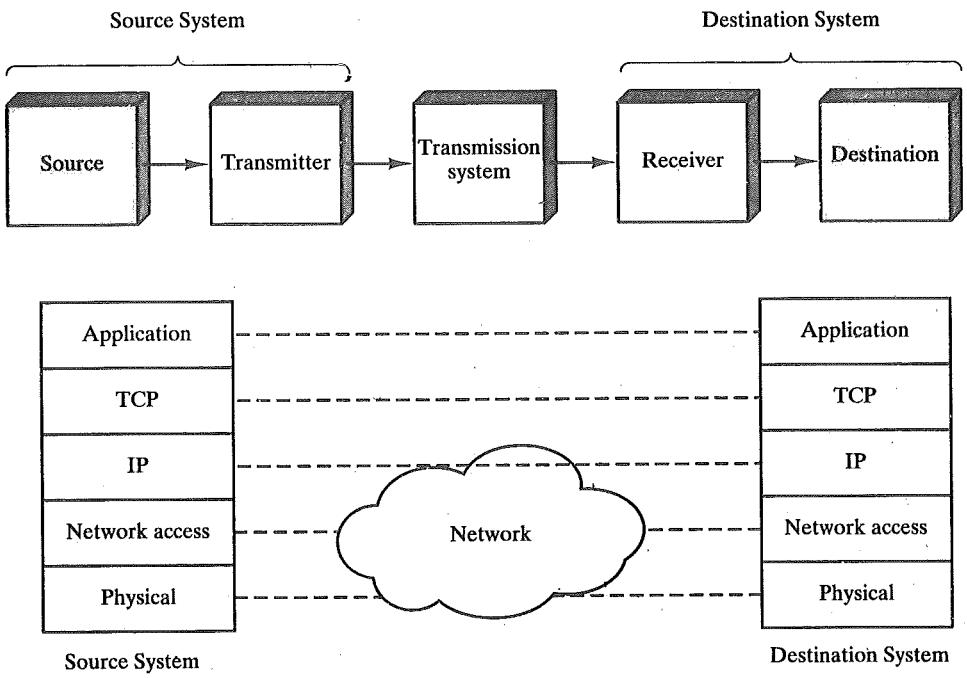


FIGURE 1.9 Protocol architecture model.

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

Figure 1.10 illustrates the OSI model and provides a brief definition of the functions performed at each layer. The intent of the OSI model is that protocols be developed to perform the functions of each layer.

The designers of OSI assumed that this model and the protocols developed within this model would come to dominate computer communications, eventually replacing proprietary protocol implementations and rival multivendor models such as TCP/IP. This has not happened. Although many useful protocols have been developed in the context of OSI, the overall seven-layer model has not flourished. Instead, it is the TCP/IP architecture that has come to dominate. Thus, our emphasis in this book will be on TCP/IP.

Application Provides access to the OSI environment for users and also provides distributed information services.
Presentation Provides independence to the application processes from differences in data representation (syntax).
Session Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications.
Transport Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.
Network Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections.
Data Link Provides for the reliable transfer of information across the physical link; sends blocks of data (frames) with the necessary synchronization, error control, and flow control.
Physical Concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium.

FIGURE 1.10 The OSI layers.

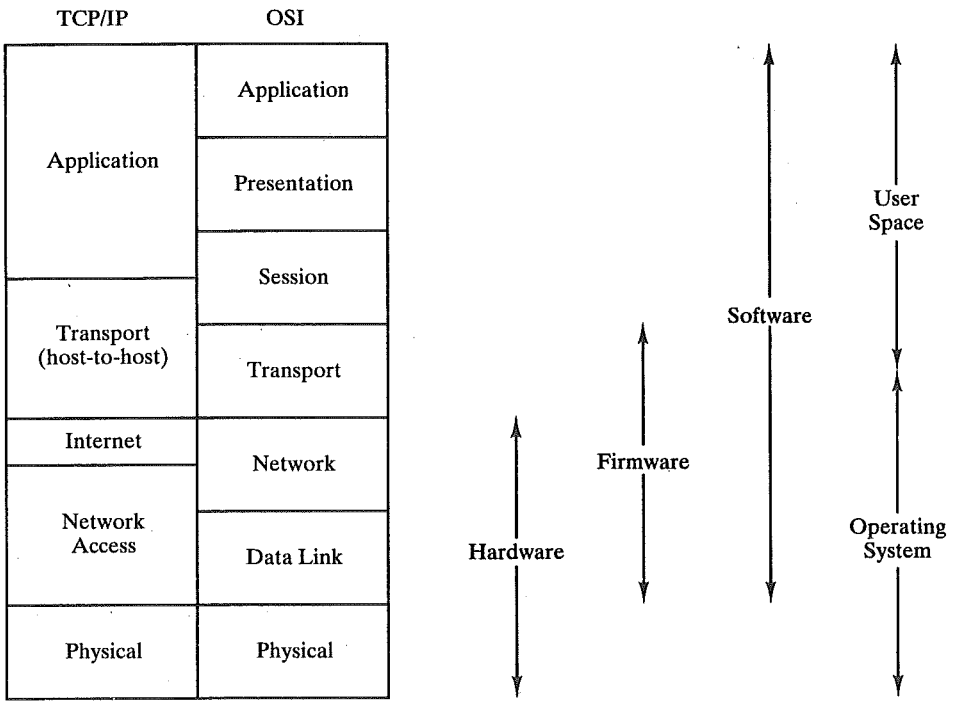


FIGURE 1.11 Protocol architectures.

Figure 1.11 illustrates the layers of the TCP/IP and OSI architectures, showing roughly the correspondence in functionality between the two. The figure also suggests common means of implementing the various layers.

1.5 STANDARDS

It has long been accepted in the communications industry that standards are required to govern the physical, electrical, and procedural characteristics of communication equipment. In the past, this view has not been embraced by the computer industry. Whereas communication-equipment vendors recognize that their equipment will generally interface to and communicate with other vendors' equipment, computer vendors have traditionally attempted to lock their customers into proprietary equipment; the proliferation of computers and distributed processing has made that an untenable position. Computers from different vendors must communicate with each other and, with the ongoing evolution of protocol standards, customers will no longer accept special-purpose protocol-conversion software development. The result is that standards now permeate all of the areas of technology discussed in this book.

Throughout the book we will describe the most important standards that are in use or that are being developed for various aspects of data and computer communications. Appendix 1A looks at the key organizations involved with the development of standards.

There are a number of advantages and disadvantages to the standards-making process. We list here the most striking ones. The principal advantages of standards are the following:

- A standard assures that there will be a large market for a particular piece of equipment or software. This encourages mass production and, in some cases, the use of large-scale-integration (LSI) or very-large-scale-integration (VLSI) techniques, resulting in lower costs.
- A standard allows products from multiple vendors to communicate, giving the purchaser more flexibility in equipment selection and use.

The principal disadvantages are these:

- A standard tends to freeze the technology. By the time a standard is developed, subjected to review and compromise, and promulgated, more efficient techniques are possible.
- There are multiple standards for the same thing. This is not a disadvantage of standards per se, but of the current way things are done. Fortunately, in recent years the various standards-making organizations have begun to cooperate more closely. Nevertheless, there are still areas where multiple conflicting standards exist.

1.6 OUTLINE OF THE BOOK

This chapter, of course, serves as an introduction to the entire book. A brief synopsis of the remaining chapters follows.

Data Transmission

The principles of data transmission underlie all of the concepts and techniques presented in this book. To understand the need for encoding, multiplexing, switching, error control, and so on, the reader must understand the behavior of data signals propagated through a transmission medium. Chapter 2 provides an understanding of the distinction between digital and analog data and digital and analog transmission. Concepts of attenuation and noise are also examined.

Transmission Media

Transmission media can be classified as either guided or wireless. The most commonly-used guided transmission media are twisted pair, coaxial cable, and optical

fiber. Wireless techniques include terrestrial and satellite microwave, broadcast radio, and infrared. Chapter 3 covers all of these topics.

Data Encoding

Data come in both analog (continuous) and digital (discrete) form. For transmission, input data must be encoded as an electrical signal that is tailored to the characteristics of the transmission medium. Both analog and digital data can be represented by either analog or digital signals; each of the four cases is discussed in Chapter 4. This chapter also covers spread-spectrum techniques.

The Data Communications Interface

In Chapter 5 the emphasis shifts from data transmission to data communications. For two devices linked by a transmission medium to exchange digital data, a high degree of cooperation is required. Typically, data are transmitted one bit at a time over the medium. The timing (rate, duration, spacing) of these bits must be the same for transmitter and receiver. Two common communication techniques—asynchronous and synchronous—are explored. This chapter also looks at transmission line interfaces. Typically, digital data devices do not attach to and signal across a transmission medium directly. Rather, this process is mediated through a standardized interface.

Data Link Control

True cooperative exchange of digital data between two devices requires some form of data link control. Chapter 6 examines the fundamental techniques common to all data link control protocols including flow control and error detection and correction, and then examines the most commonly used protocol, HDLC.

Multiplexing

Transmission facilities are, by and large, expensive. It is often the case that two communication stations will not utilize the full capacity of a data link. For efficiency, it should be possible to share that capacity. The generic term for such sharing is multiplexing.

Chapter 7 concentrates on the three most common types of multiplexing techniques. The first, frequency-division multiplexing (FDM), is the most widespread and is familiar to anyone who has ever used a radio or television set. The second is a particular case of time-division multiplexing (TDM), often known as synchronous TDM. This is commonly used for multiplexing digitized voice streams. The third type is another form of TDM that is more complex but potentially more efficient than synchronous TDM; it is referred to as statistical or asynchronous TDM.

Circuit Switching

Any treatment of the technology and architecture of circuit-switched networks must of necessity focus on the internal operation of a single switch. This is in con-

trast to packet-switched networks, which are best explained by the collective behavior of the set of switches that make up a network. Thus, Chapter 8 begins by examining digital-switching concepts, including space- and time-division switching. Then, the concepts of a multinode circuit-switched network are discussed; here, we are primarily concerned with the topics of routing and control signaling.

Packet Switching

There are two main technical problems associated with a packet-switched network, and each is examined in Chapter 9:

- **Routing.** Because the source and destination stations are not directly connected, the network must route each packet, from node to node, through the network.
- **Congestion control.** The amount of traffic entering and transiting the network must be regulated for efficient, stable, and fair performance.

The key design issues in both of these areas are presented and analyzed; the discussion is supported by examples from specific networks. In addition, a key packet-switching interface standard, X.25, is described.

Frame Relay

Chapter 10 examines the most important innovation to come out of the work on ISDN: frame relay. Frame relay provides a more efficient means of supporting packet switching than X.25 and is enjoying widespread use, not only in ISDN but in other networking contexts. This chapter looks at the data-transfer protocol and call-control protocol for frame relay and also looks at the related data link control protocol, LAPF.

A critical component for frame relay is congestion control. The chapter explains the nature of congestion in frame relay networks and both the importance and difficulty of controlling congestion. The chapter then describes a range of congestion control techniques that have been specified for use in frame relay networks.

Asynchronous Transfer Mode (ATM)

Chapter 11 focuses on the transmission technology that is the foundation of broadband ISDN: asynchronous transfer mode (ATM). As with frame relay, ATM is finding widespread application beyond its use as part of broadband. This chapter begins with a description of the ATM protocol and format. Then the physical layer issues relating to the transmission of ATM cells and the ATM Adaptation Layer (AAL) are discussed.

Again, as with frame relay, congestion control is a vital component of ATM. This area, referred to as ATM traffic and congestion control, is one of the most complex aspects of ATM and is the subject of intensive ongoing research. This chapter surveys those techniques that have been accepted as having broad utility in

ATM environments.

LAN Technology

The essential technology underlying all forms of local area networks comprises topology, transmission medium, and medium access control technique. Chapter 12 examines the first two of these elements. Four topologies are in common use: bus, tree, ring, star. The most common transmission media for local networking are twisted pair (unshielded and shielded), coaxial cable (baseband and broadband), and optical fiber. These topologies and transmission media are discussed, and the most promising combinations are described.

LAN Systems

Chapter 13 looks in detail at the topologies, transmission media, and MAC protocols of the most important LAN systems in current use; all of these have been defined in standards documents. The discussion opens with what might be called traditional LANs, which typically operate at data rates of up to 10 Mbps and which have been in use for over a decade. These include Ethernet and related LANs and two token-passing schemes, token ring and FDDI (fiber distributed data interface). Then, more recent high-speed LAN systems are examined, including ATM LANs. Finally, the chapter looks at wireless LANs.

Bridges

The increasing deployment of LANs has led to an increased need to interconnect LANs with each other and with wide-area networks. Chapter 14 focuses on a key device used in interconnection LANs: the bridges. Bridge operation involves two types of protocols: protocols for forwarding packets and protocols for exchanging routing information.

This chapter also returns to the topic of ATM LANs to look at the important concept of ATM LAN emulation, which relates to connecting other types of LANs to ATM networks.

Protocols and Architecture

Chapter 15 introduces the subject of protocol architecture and motivates the need for a layered architecture with protocols defined at each layer. The concept of protocol is defined, and the important features of protocols are discussed.

The two most important communications architectures are introduced in this chapter. The open systems interconnection (OSI) model is described in some detail. Next, the TCP/IP model is examined. Although the OSI model is almost universally accepted as the framework for discourse in this area, it is the TCP/IP protocol suite that is the basis for most commercially available interoperable products.

Internetworking

With the proliferation of networks, internetworking facilities have become essential components of network design. Chapter 16 begins with an examination of the requirements for an internetworking facility and the various design approaches that can be taken to satisfy those requirements. The remainder of the chapter explores the use of routers for internetworking. The internet protocol (IP) and the new IPv6, also known as IPng, are examined. Various routing protocols are also described, including the widely used OSPF and BGP.

Transport Protocols

The transport protocol is the keystone of the whole concept of a computer communications architecture. It can also be one of the most complex of protocols. Chapter 17 examines in detail transport protocol mechanisms and then introduces two important examples: TCP and UDP.

Network Security

Network security has become increasingly important with the growth in the number and importance of networks. Chapter 18 provides a survey of security techniques and services. The chapter begins with a look at encryption techniques for insuring privacy, which include the use of conventional and public-key encryption. Then, the area of authentication and digital signatures is explored. The two most important encryption algorithms, DES and RSA, are examined, as well as MD5, a one-way hash function important in a number of security applications.

Distributed Applications

The purpose of a communications architecture is to support distributed applications. Chapter 19 examines three of the most important of these applications; in each case, general principles are discussed and are followed by a specific example. The applications discussed are network management, world-wide web (WWW) exchanges, and electronic mail. The corresponding examples are SNMPv2, HTTP, and SMTP/MIME. Before getting to these examples, the chapter opens with an examination of Abstract Syntax Notation One (ASN.1), which is the standardized language for defining distributed applications.

ISDN and Broadband ISDN

The integrated-services digital network (ISDN) is a projected worldwide public telecommunications network that is designed to service a variety of user needs. Broadband ISDN is an enhancement of ISDN that can support very high data rates. Appendix A looks at the architecture, design principles, and standards for ISDN and broadband ISDN.

1A APPENDIX

STANDARDS ORGANIZATIONS

THROUGHOUT THIS BOOK, we describe the most important standards in use or being developed for various aspects of data and computer communications. Various organizations have been involved in the development or promotion of these standards. This appendix provides a brief description of the most important (in the current context) of these organizations:

- IETF
- ISO
- ITU-T

Internet Standards and the IETF

Many of the protocols that make up the TCP/IP protocol suite have been standardized or are in the process of standardization. By universal agreement, an organization known as the Internet Architecture Board (IAB) is responsible for the development and publication of these standards, which are published in a series of documents called Requests for Comments (RFCs).

This section provides a brief description of the way in which standards for the TCP/IP protocol suite are developed.

The Internet and Internet Standards

The Internet is a large collection of interconnected networks, all of which use the TCP/IP protocol suite. The Internet began with the development of ARPANET and the subsequent support by the Defense Advanced Research Projects Agency (DARPA) for the development of additional networks to support military users and government contractors.

The IAB is the coordinating committee for Internet design, engineering, and management. Areas covered include the operation of the Internet itself and the standardization of protocols used by end systems on the Internet for interoperability. The IAB has two principle subsidiary task forces:

- Internet Engineering Task Force (IETF)
- Internet Research Task Force (IRTF)

The actual work of these task forces is carried out by working groups. Membership in a working group is voluntary; any interested party may participate.

It is the IETF that is responsible for publishing the RFCs. The RFCs are the working notes of the Internet research and development community. A document in this series may be on essentially any topic related to computer communications, and may be anything from a meeting report to the specification of a standard.

The final decision of which RFCs become Internet standards is made by the IAB, on the recommendation of the IETF. To become a standard, a specification must meet the following criteria:

- Be stable and well-understood
- Be technically competent
- Have multiple, independent, and interoperable implementations with operational experience

- Enjoy significant public support
- Be recognizably useful in some or all parts of the Internet

The key difference between these criteria and those used for international standards is the emphasis here on operational experience.

The Standardization Process

Figure 1.12 shows the series of steps, called the *standards track*, that a specification goes through to become a standard. The steps involve increasing amounts of scrutiny and testing. At each step, the IETF must make a recommendation for advancement of the protocol, and the IAB must ratify it.

The white boxes in the diagram represent temporary states, which should be occupied for the minimum practical time. However, a document must remain a proposed standard for at least six months and a draft standard for at least four months to allow time for review and comment. The gray boxes represent long-term states that may be occupied for years.

A protocol or other specification that is not considered ready for standardization may be published as an experimental RFC. After further work, the specification may be resubmitted. If the specification is generally stable, has resolved known design choices, is believed to be well-understood, has received significant community review, and appears to enjoy enough community interest to be considered valuable, then the RFC will be designated a proposed standard.

For a specification to be advanced to draft-standard status, there must be at least two independent and interoperable implementations from which adequate operational experience has been obtained.

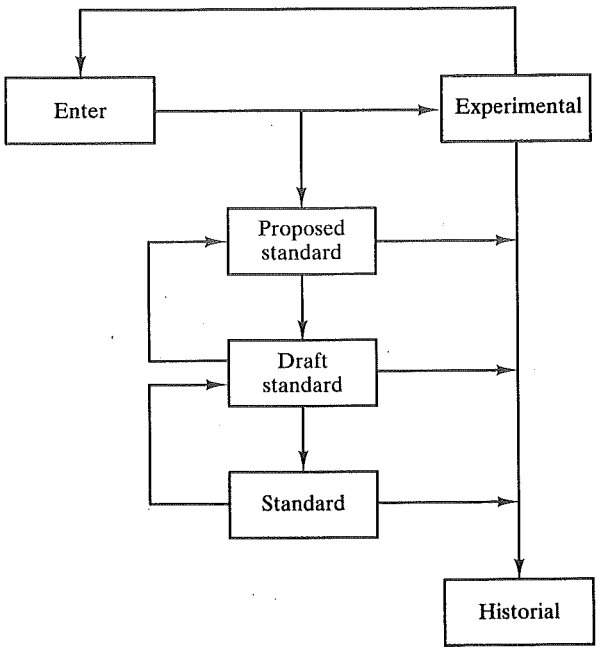


FIGURE 1.12 Standards track diagram.

After significant implementation and operational experience has been obtained, a specification may be elevated to standard. At this point, the specification is assigned an STD number as well as an RFC number.

Finally, when a protocol becomes obsolete, it is assigned to the historic state.

The International Organization for Standardization (ISO)

ISO is an international agency for the development of standards on a wide range of subjects. It is a voluntary, nontreaty organization whose members are designated standards bodies of participating nations, plus nonvoting observer organizations. Although ISO is not a governmental body, more than 70 percent of ISO member bodies are governmental standards institutions or organizations incorporated by public law. Most of the remainder have close links with the public administrations in their own countries. The United States member body is the American National Standards Institute.

ISO was founded in 1946 and has issued more than 5000 standards in a broad range of areas. Its purpose is to promote the development of standardization and related activities to facilitate international exchange of goods and services and to develop cooperation in the sphere of intellectual, scientific, technological, and economic activity. Standards have been issued to cover everything from screw threads to solar energy. One important area of standardization deals with the open systems interconnection (OSI) communications architecture and the standards at each layer of the OSI architecture.

In the areas of interest in this book, ISO standards are actually developed in a joint effort with another standards body, the International Electrotechnical Commission (IEC). IEC is primarily concerned with electrical and electronic engineering standards. In the area of information technology, the interests of the two groups overlap, with IEC emphasizing hardware and ISO focusing on software. In 1987, the two groups formed the Joint Technical Committee 1 (JTC 1). This committee has the responsibility of developing the documents that ultimately become ISO (and IEC) standards in the area of information technology.

The development of an ISO standard from first proposal to actual publication of the standard follows a seven-step process. The objective is to ensure that the final result is acceptable to as many countries as possible. The steps are briefly described here. (Time limits are the minimum time in which voting could be accomplished, and amendments require extended time.)

1. A new work item is assigned to the appropriate technical committee, and within that technical committee, to the appropriate working group. The working group prepares the technical specifications for the proposed standard and publishes these as a draft proposal (DP). The DP is circulated among interested members for balloting and technical comment. At least three months are allowed, and there may be iterations. When there is substantial agreement, the DP is sent to the administrative arm of ISO, known as the Central Secretariat.
2. The DP is registered at the Central Secretariat within two months of its final approval by the technical committee.
3. The Central Secretariat edits the document to ensure conformity with ISO practices; no technical changes are made. The edited document is then issued as a draft international standard (DIS).
4. The DIS is circulated for a six-month balloting period. For approval, the DIS must receive a majority approval by the technical committee members and 75 percent approval of all voting members. Revisions may occur to resolve any negative vote. If more than two negative votes remain, it is unlikely that the DIS will be published as a final standard.
5. The approved, possibly revised, DIS is returned within three months to the Central Secretariat for submission to the ISO Council, which acts as the board of directors of ISO.

6. The DIS is accepted by the Council as an international standard (IS).
7. The IS is published by ISO.

As can be seen, the process of issuing a standard is a slow one. Certainly, it would be desirable to issue standards as quickly as the technical details can be worked out, but ISO must ensure that the standard will receive widespread support.

ITU Telecommunications Standardization Sector

The ITU Telecommunications Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union (ITU), which is itself a United Nations specialized agency. Hence, the members of ITU-T are governments. The U.S. representation is housed in the Department of State. The charter of the ITU is that it "is responsible for studying technical, operating, and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis." Its primary objective is to standardize, to the extent necessary, techniques and operations in telecommunications to achieve end-to-end compatibility of international telecommunication connections, regardless of the countries of origin and destination.

The ITU-T was created on March 1, 1993, as one consequence of a reform process within the ITU. It replaces the International Telegraph and Telephone Consultative Committee (CCITT), which had essentially the same charter and objectives as the new ITU-T.

ITU-T is organized into 15 study groups that prepare Recommendations:

1. Service Description
2. Network Operation
3. Tariff and Accounting Principles
4. Network Maintenance
5. Protection Against Electromagnetic Environment Effects
6. Outside Plant
7. Data Network and Open Systems Communications
8. Terminal Equipment and Protocols for Telematic Services
9. Television and Sound Transmission
10. Languages for Telecommunication Applications
11. Switching and Signalling
12. End-to-End Transmission Performance
13. General Network Aspects
14. Modems and Transmission Techniques for Data, Telegraph, and Telematic Services
15. Transmission Systems and Equipment

Work within ITU-T is conducted in four-year cycles. Every four years, a World Telecommunications Standardization Conference is held. The work program for the next four years is established at the assembly in the form of questions submitted by the various study groups, based on requests made to the study groups by their members. The conference assesses the questions, reviews the scope of the study groups, creates new or abolishes existing study groups, and allocates questions to these groups.

Based on these questions, each study group prepares draft Recommendations. A draft Recommendation may be submitted to the next conference, four years hence, for approval. Increasingly, however, Recommendations are approved when they are ready, without having to wait for the end of the four-year Study Period. This accelerated procedure was adopted after the study period that ended in 1988. Thus, 1988 was the last time that a large batch of documents was published at one time as a set of Recommendations.

1B APPENDIX

INTERNET RESOURCES

THERE ARE A number of resources available on the Internet for keeping up with developments in this field.

USENET Newsgroups

A number of USENET newsgroups are devoted to some aspect of data communications and networking. As with virtually all USENET groups, there is a high noise-to-signal ratio, but it is worth experimenting to see if any meet your needs. Here is a sample:

- comp.dcom.lans, comp.dcom.lans.misc: General discussions of LANs.
- comp.std.wireless: General discussion of wireless networks, including wireless LANs.
- comp.security.misc: Computer security and encryption.
- comp.dcom.cell-relay: Covers ATM and ATM LANs.
- comp.dcom.frame-relay: Covers frame-relay networks.
- comp.dcom.net-management: Discussion of network-management applications, protocols, and standards.
- comp.protocols.tcp-ip: The TCP/IP protocol suite.

Web Sites for This Book

A special web page has been set up for this book at http://www.shore.net/~ws/DCC5e.html The site includes the following:

- Links to other web sites, including the sites listed in this book, provide a gateway to relevant resources on the web.
- Links to papers and reports available via the Internet provide additional, up-to-date material for study.
- We also hope to include to links to home pages for courses based on the book; these pages may be useful to other instructors in providing ideas about how to structure the course.
- Additional problems, exercises, and other activities for classroom use are also planned.

As soon as any typos or other errors are discovered, an errata list for this book will be available at http://www.shore.net/~ws/welcome.html. The file will be updated as needed. Please email any errors that you spot to ws@shore.net. Errata sheets for other books are at the same web site, as well as discount ordering information for the books.

Other Web Sites

There are numerous web sites that provide some sort of information related to the topics of this book. Here is a sample:

- http://www.soc.hawaii.edu/con/com-resources.html: Information and links to resources about data communications and networking.
- http://www.internic.net/ds/dspg01.html: Maintains archives that relate to the Internet and IETF activities. Includes keyword-indexed library of RFCs and draft documents as well as many other documents related to the Internet and related protocols.

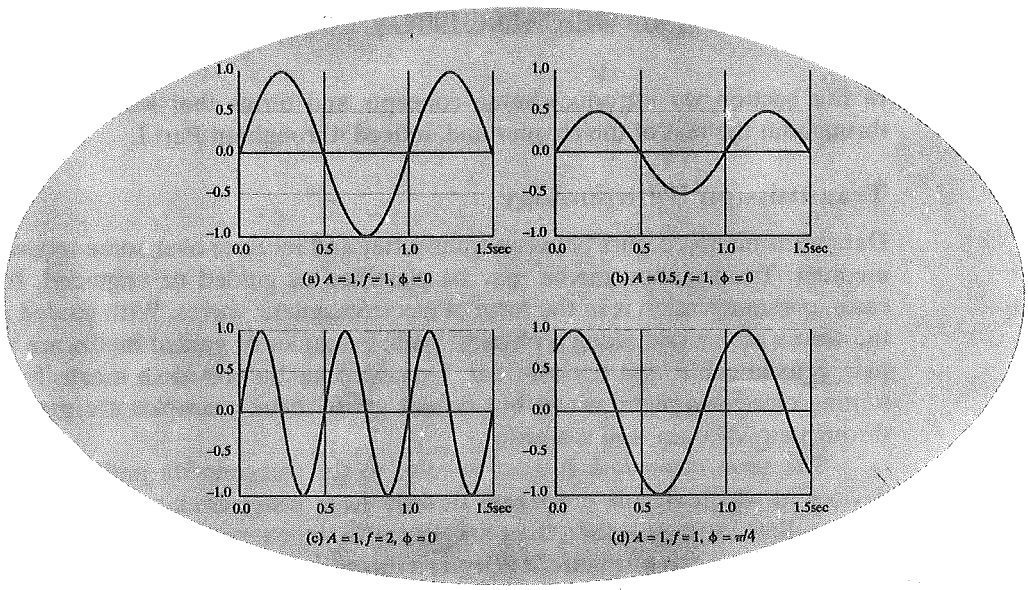
- http://www.ronin.com/SBA: Links to over 1500 hardware and software vendors who currently have WWW sites, as well as a list of thousands of computer and networking companies in a Phone Directory.
- http://liinwww.ira.uka.de/bibliography/index.html: The Computer Science Bibliography Collection, a collection of hundreds of bibliographies with hundreds of thousands of references.

In subsequent chapters, pointers to more specific web sites can be found in the "Recommended Reading" section.

PART ONE Data Communications

CHAPTER 2

DATA TRANSMISSION



The successful transmission of data depends principally on two factors: the quality of the signal being transmitted and the characteristics of the transmission medium. The objective of this chapter and the next is to provide the reader with an intuitive feeling for the nature of these two factors.

The first section presents some concepts and terms from the field of electrical engineering; this should provide sufficient background for the remainder of the chapter. Section 2.2 clarifies the use of the terms *analog* and *digital*. Either analog or digital data may be transmitted using either analog or digital signals. Furthermore, it is common for intermediate processing to be performed between source and destination, and this processing has either an analog or digital character.

Section 2.3 looks at the various impairments that may introduce errors into the data during transmission. The chief impairments are attenuation, delay distortion, and the various forms of noise.

2.1 CONCEPTS AND TERMINOLOGY

In this section we introduce some concepts and terms that will be referred to throughout the rest of the chapter and, indeed, throughout Part I.

Transmission Terminology

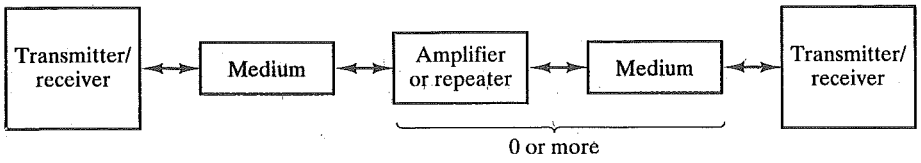
Data transmission occurs between transmitter and receiver over some transmission medium. Transmission media may be classified as guided or unguided. In both cases, communication is in the form of electromagnetic waves. With guided media, the waves are guided along a physical path; examples of guided media are twisted pair, coaxial cable, and optical fiber. Unguided media provide a means for transmitting electromagnetic waves but do not guide them; examples are propagation through air, vacuum, and sea water.

The term *direct link* is used to refer to the transmission path between two devices in which signals propagate directly from transmitter to receiver with no intermediate devices, other than amplifiers or repeaters used to increase signal strength. Both parts of Figure 2.1 depict a direct link. Note that this term can apply to both guided and unguided media.

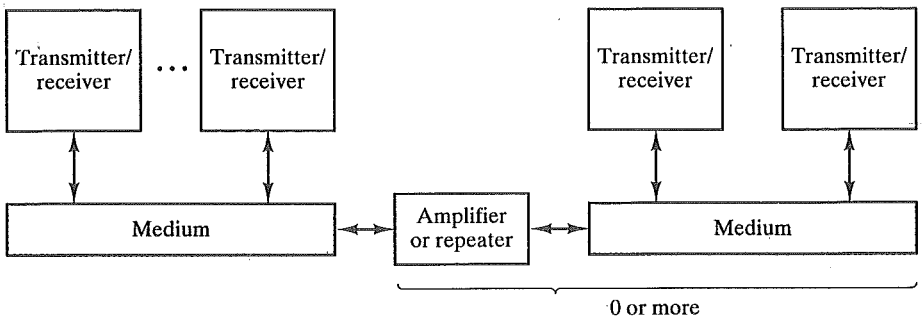
A guided transmission medium is point-to-point if, first, it provides a direct link between two devices and, second, those are the only two devices sharing the medium (Figure 2.1a). In a multipoint guided configuration, more than two devices share the same medium (Figure 2.1b).

A transmission may be simplex, half-duplex, or full-duplex. In simplex transmission, signals are transmitted in only one direction; one station is the transmitter and the other is the receiver. In half-duplex operation, both stations may transmit, but only one at a time. In full-duplex operation, both stations may transmit simultaneously. In the latter case, the medium is carrying signals in both directions at the same time. How this can be is explained in due course.

We should note that the definitions just given are the ones in common use in the United States (ANSI definitions). In Europe (ITU-T definitions), the term



(a) Point-to-point



(b) Multipoint

FIGURE 2.1 Guided transmission configurations.

"simplex" is used to correspond to half-duplex, as defined above, and "duplex" is used to correspond to full-duplex, as also defined above.

Frequency, Spectrum, and Bandwidth

In this book, we are concerned with electromagnetic signals, used as a means to transmit data. At point 3 in Figure 1.1, a signal is generated by the transmitter and transmitted over a medium. The signal is a function of time, but it can also be expressed as a function of frequency; that is, the signal consists of components of different frequencies. It turns out that the *frequency-domain* view of a signal is far more important to an understanding of data transmission than a *time-domain* view. Both views are introduced here.

Time-Domain Concepts

Viewed as a function of time, an electromagnetic signal can be either continuous or discrete. A continuous signal is one in which the signal intensity varies in a smooth fashion over time. In other words, there are no breaks or discontinuities in the signal.[1] A discrete signal is one in which the signal intensity maintains a constant level for some period of time and then changes to another constant level. Figure 2.2 shows examples of both kinds of signals. The continuous signal might represent speech, and the discrete signal might represent binary 1s and 0s.

The simplest sort of signal is a *periodic signal*, in which the same signal pattern repeats over time. Figure 2.3 shows an example of a periodic analog signal (sine

[1] A mathematical definition: A signal $s(t)$ is continuous if $\lim_{t \to a} s(t) = s(a)$ for all a.

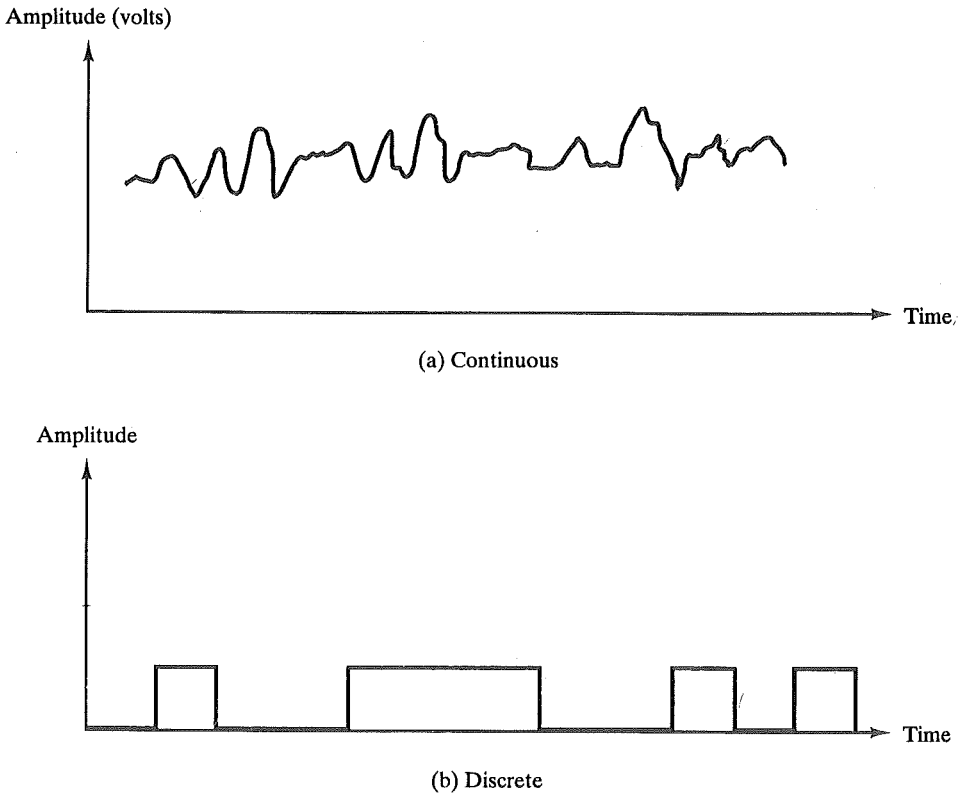


FIGURE 2.2 Continuous and discrete signals.

wave) and a periodic digital signal (square wave). Mathematically, a signal $s(t)$ is defined to be periodic if and only if

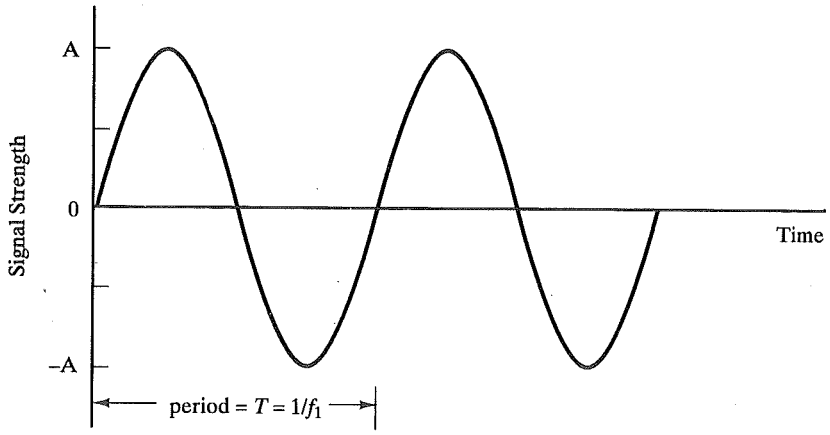
$$s(t + T) = s(t) \qquad -\infty < t < +\infty$$

where the constant T is the period of the signal. (T is the smallest value that satisfies the equation.) Otherwise, a signal is aperiodic.

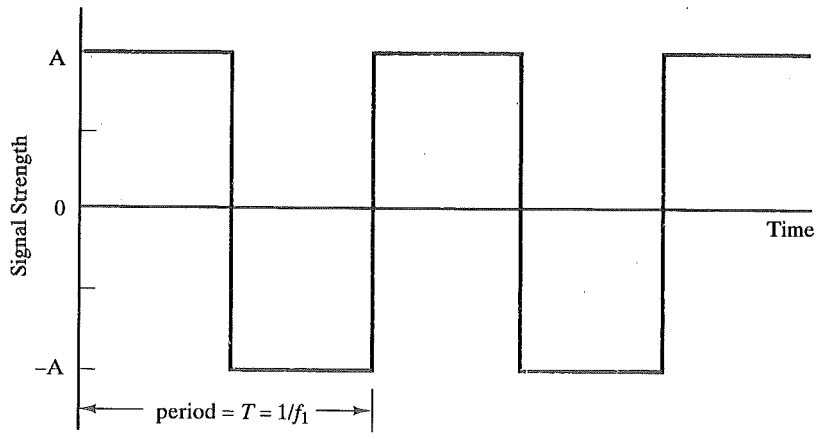
The sine wave is the fundamental continuous signal. A general sine wave can be represented by three parameters: amplitude (A), frequency (f), and phase (ϕ). The *amplitude* is the peak value or strength of the signal over time; typically, this value is measured in volts or watts. The *frequency* is the rate (in cycles per second, or Hertz (Hz)) at which the signal repeats. An equivalent parameter is the *period* (T) of a signal, which is the amount of time it takes for one repetition; therefore, $T = 1/f$. *Phase* is a measure of the relative position in time within a single period of a signal, as illustrated below.

The general sine wave can be written

$$s(t) = A \sin(2\pi ft + \phi)$$



(a) Sine wave



(b) Square wave

FIGURE 2.3 Example of periodic signals.

Figure 2.4 shows the effect of varying each of the three parameters. In part (a) of the figure, the frequency is 1 Hz; thus, the period is $T = 1$ second. Part (b) has the same frequency and phase but an amplitude of 1/2. In part (c), we have $f = 2$, which is equivalent to $T = 1/2$. Finally, part (d) shows the effect of a phase shift of $\pi/4$ radians, which is 45 degrees (2π radians = 360° = 1 period).

In Figure 2.4, the horizontal axis is time; the graphs display the value of a signal at a given point in space as a function of time. These same graphs, with a change of scale, can apply with horizontal axes in space. In this case, the graphs display the value of a signal at a given point in time as a function of distance. For example, for a sinusoidal transmission (say an electromagnetic radio wave some distance from a radio antenna, or sound some distance from a loudspeaker), at a particular instant of time, the intensity of the signal varies in a sinusoidal way as a function of distance from the source.

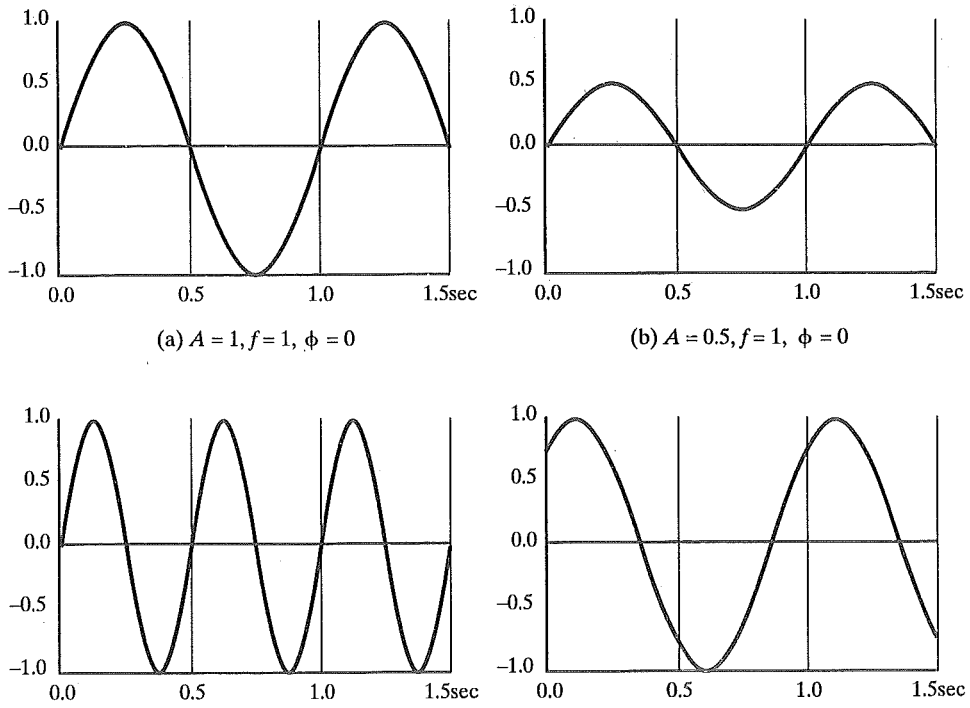


FIGURE 2.4 $A \sin(2\pi ft + \phi)$.

There are two simple relationships between the two sine waves, one in time and one in space. Define the *wavelength*, λ, of a signal as the distance occupied by a single cycle, or, put another way, as the distance between two points of corresponding phase of two consecutive cycles. Assume that the signal is traveling with a velocity v. Then the wavelength is related to the period as follows: $\lambda = vT$. Equivalently, $\lambda f = v$. Of particular relevance to this discussion is the case where $v = c$, the speed of light in free space, which is 3×10^8 m/s.

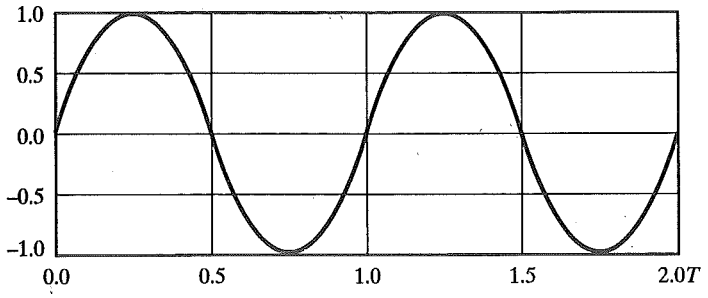
Frequency Domain Concepts

In practice, an electromagnetic signal will be made up of many frequencies. For example, the signal

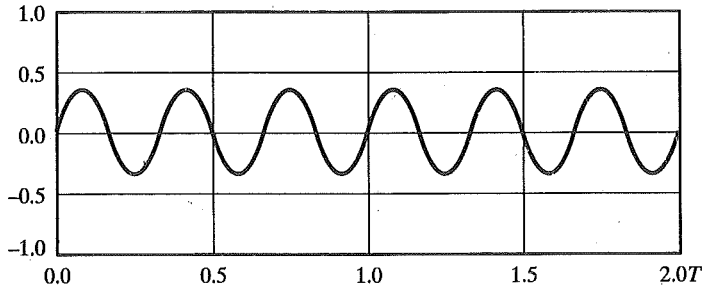
$$s(t) = \sin(2\pi f_1 t) + \frac{1}{3}\sin(2\pi(3f_1)t)$$

is shown in Figure 2.5. The components of this signal are just sine waves of frequencies f_1 and $3f_1$; parts a and b of the figure show these individual components. There are several interesting points that can be made about this figure:

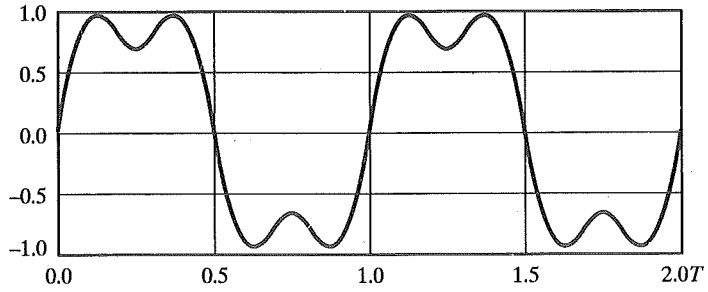
- The second frequency is an integer multiple of the first frequency. When all of the frequency components of a signal are integer multiples of one frequency, the latter frequency is referred to as the fundamental frequency.



(a) sin $(2\pi f_1 t)$



(b) 1/3 sin $(2\pi(3f_1)t)$

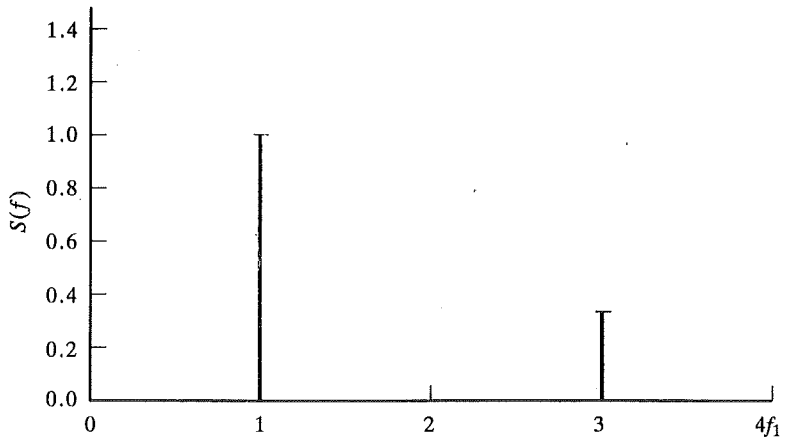


(c) sin $(2\pi f_1 t)$ + 1/3 sin $(2\pi(3f_1)t)$

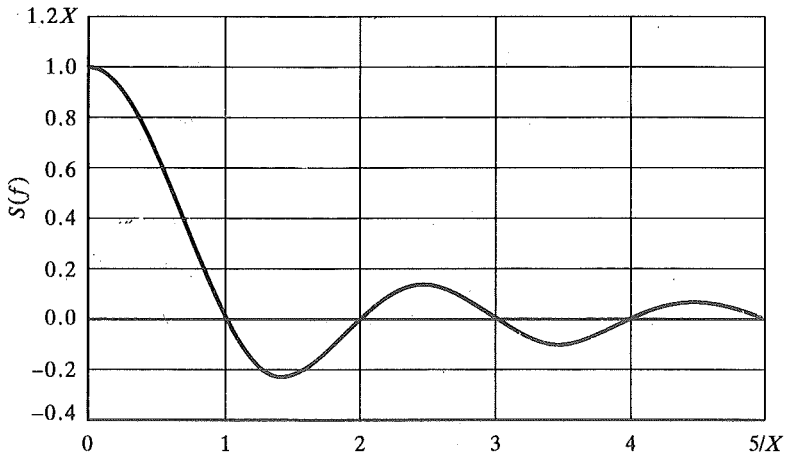
FIGURE 2.5 Addition of frequency components ($T = 1/f_1$).

- The period of the total signal is equal to the period of the fundamental frequency. The period of the component sin $(2\pi f_1 t)$is $T = 1/f_1$, and the period of $s(t)$ is also T, as can be seen from Figure 2.5c.

It can be shown, using a discipline known as Fourier analysis, that any signal is made up of components at various frequencies, in which each component is a sinusoid. This result is of tremendous importance, because the effects of various transmission media on a signal can be expressed in terms of frequencies, as is discussed later in this chapter. For the interested reader, the subject of Fourier analysis is introduced in Appendix 2A at the end of this chapter.



(a) $s(t) = \sin(2\pi f_1 t) + 1/3 \sin(3(2\pi f_1 t))$

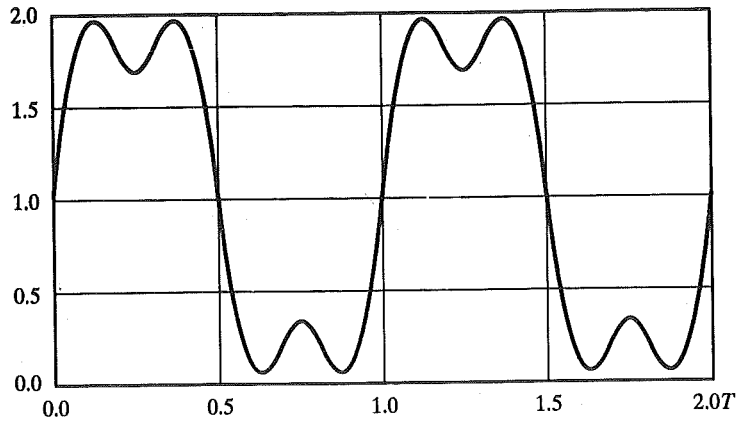


(b) $s(t) = 1 - X/2 \le t \le X/2$

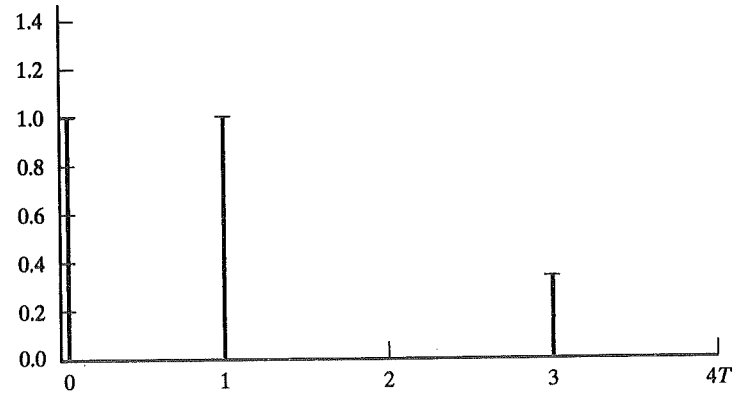
FIGURE 2.6 Frequency-domain representations.

So, we can say that for each signal, there is a time-domain function $s(t)$ that specifies the amplitude of the signal at each instant in time. Similarly, there is a frequency-domain function $S(f)$ that specifies the constituent frequencies of the signal. Figure 2.6a shows the frequency-domain function for the signal in Figure 2.5c. Note that, in this case, $S(f)$ is discrete. Figure 2.6b shows the frequency domain function for a single square pulse that has the value 1 between $-X/2$ and $X/2$, and is 0 elsewhere. Note that in this case $S(f)$ is continuous, and that it has nonzero values indefinitely, although the magnitude of the frequency components becomes smaller for larger f. These characteristics are common for real signals.

The *spectrum* of a signal is the range of frequencies that it contains. For the signal in Figure 2.5c, the spectrum extends from f_1 to $3f_1$. The *absolute bandwidth* of a signal is the width of the spectrum. In the case of Figure 2.5c, the bandwidth is $2f_1$. Many signals, such as that of Figure 2.6b, have an infinite bandwidth. However,



(a) $s(t) = 1 + \sin + 1/3 \sin (2\pi(3f_1)t)$



(b) $S(f)$

FIGURE 2.7 Signal with dc component.

most of the energy in the signal is contained in a relatively narrow band of frequencies. This band is referred to as the *effective bandwidth*, or just *bandwidth*.

One final term to define is *dc component*. If a signal includes a component of zero frequency, that component is a direct current (dc) or constant component. For example, Figure 2.7 shows the result of adding a dc component to the signal of Figure 2.6. With no dc component, a signal has an average amplitude of zero, as seen in the time domain. With a dc component, it has a frequency term at $f = 0$ and a nonzero average amplitude.

Relationship Between Data Rate and Bandwidth

The concept of effective bandwidth is a somewhat fuzzy one. We have said that it is the band within which most of the signal energy is confined. The term "most" in this context is somewhat arbitrary. The important issue here is that, although a given waveform may contain frequencies over a very broad range, as a practical matter

any transmission medium that is used will be able to accommodate only a limited band of frequencies. This, in turn, limits the data rate that can be carried on the transmission medium.

To try to explain these relationships, consider the square wave of Figure 2.3b. Suppose that we let a positive pulse represent binary 1 and a negative pulse represent binary 0. Then, the waveform represents the binary stream 1010 . . . . The duration of each pulse is $1/2f_1$; thus, the data rate is $2f_1$ bits per second (bps). What are the frequency components of this signal? To answer this question, consider again Figure 2.5. By adding together sine waves at frequencies f_1 and $3f_1$, we get a waveform that resembles the square wave. Let us continue this process by adding a sine wave of frequency $5f_1$, as shown in Figure 2.8a, and then adding a sine wave of frequency $7f_1$, as shown in Figure 2.8b. As we add additional odd multiples of f_1, suitably scaled, the resulting waveform approaches more and more closely that of a square wave.

Indeed, it can be shown that the frequency components of the square wave can be expressed as follows:

$$s(t) = A \times \sum_{k \text{ odd}, k=1}^{\infty} \frac{1}{k} \sin(2\pi k f_1 t)$$

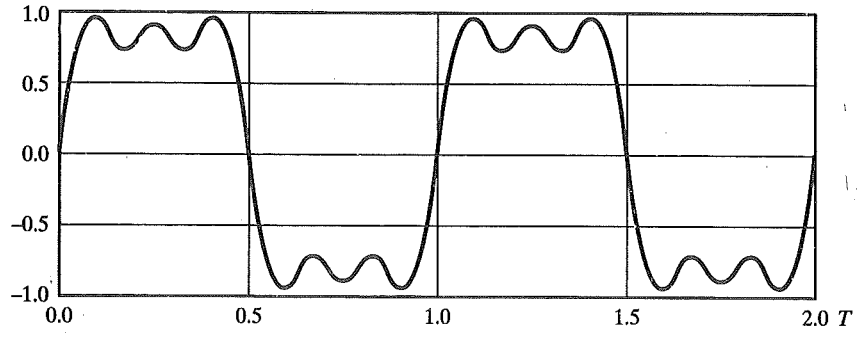
Thus, this waveform has an infinite number of frequency components and, hence, an infinite bandwidth. However, the amplitude of the kth frequency component, kf_1, is only $1/k$, so most of the energy in this waveform is in the first few frequency components. What happens if we limit the bandwidth to just the first three frequency components? We have already seen the answer, in Figure 2.8a. As we can see, the shape of the resulting waveform is reasonably close to that of the original square wave.

We can use Figures 2.5 and 2.8 to illustrate the relationship between data rate and bandwidth. Suppose that we are using a digital transmission system that is capable of transmitting signals with a bandwidth of 4 MHz. Let us attempt to transmit a sequence of alternating 1s and 0s as the square wave of Figure 2.8c. What data rate can be achieved? Let us approximate our square wave with the waveform of Figure 2.8a. Although this waveform is a "distorted" square wave, it is sufficiently close to the square wave that a receiver should be able to discriminate between a binary 0 and a binary 1. Now, if we let $f_1 = 10^6$ cycles/second = 1 MHz, then the bandwidth of the signal

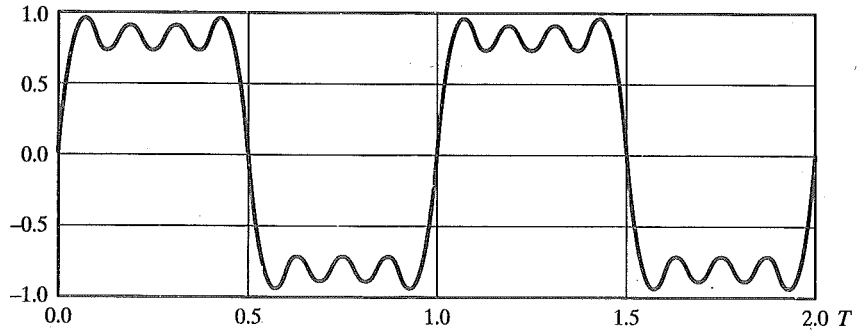
$$s(t) = \sin((2\pi \times 10^6)t) + \frac{1}{3}\sin((2\pi \times 3 \times 10^6)t) + \frac{1}{5}\sin((2\pi \times 5 \times 10^6)t)$$

is $(5 \times 10^6) - 10^6 = 4$ MHz. Note that for $f_1 = 1$ MHz, the period of the fundamental frequency is $T = 1/10^6 = 10^{-6} = 1$ μsec. Thus, if we treat this waveform as a bit string of 1s and 0s, one bit occurs every 0.5 μsec, for a data rate of $2 \times 10^6 = 2$ Mbps. Thus, for a bandwidth of 4 Mhz, a data rate of 2 Mbps is achieved.

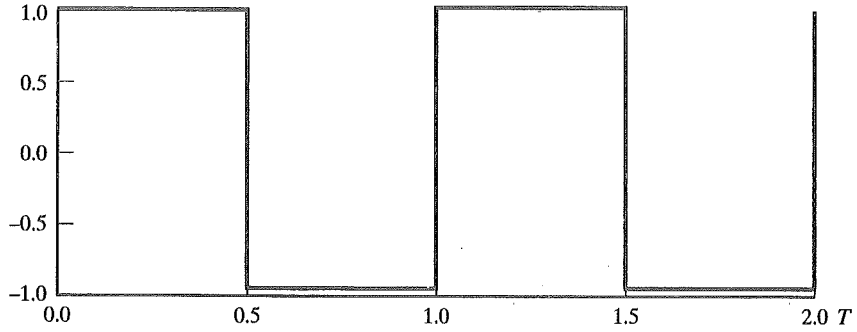
Now suppose that we have a bandwidth of 8 MHz. Let us look again at Figure 2.8a, but now with $f_1 = 2$ MHz. Using the same line of reasoning as before, the bandwidth of the signal is $(5 \times 2 \times 10^6) - (2 \times 10^6) = 8$ MHz. But in this case $T = 1/f_1 = 0.5$ μsec. As a result, one bit occurs every 0.25 μsec for a data rate of



(a) $\sin(2\pi f_1 t) + 1/3 \sin(2\pi(3f_1)t) + 1/5 \sin(2\pi(5f_1)t)$



(b) $\sin(2\pi f_1 t) + 1/3 \sin(2\pi(3f_1)t) + 1/5 \sin(2\pi(5f_1)t) + 1/7 \sin(2\pi(7f_1)t)$



(c) Σ 1/k $\sin(2\pi f_1 t)$

FIGURE 2.8 Frequency components of a square wave ($T = 1/f_1$).

4 Mbps. Thus, other things being equal, by doubling the bandwidth, we double the potential data rate.

But now suppose that the waveform in Figure 2.5c is considered adequate for approximating a square wave. That is, the difference between a positive and negative pulse in Figure 2.5c is sufficiently distinct that the waveform can be successfully used to represent a sequence of 1s and 0s. Now, let $f_1 = 2$ MHz. Using the same line

of reasoning as before, the bandwidth of the signal in Figure 2.5c is $(3 \times 2 \times 10^6) - (2 \times 10^6) = 4$ MHz. But, in this case, $T = 1/f_1 = 0.5$ μsec. As a result, one bit occurs every 0.25 μsec, for a data rate of 4 Mbps. Thus, a given bandwidth can support various data rates depending on the requirements of the receiver.

We can draw the following general conclusions from the above observations. In general, any digital waveform will have infinite bandwidth. If we attempt to transmit this waveform as a signal over any medium, the nature of the medium will limit the bandwidth that can be transmitted. Furthermore, for any given medium, the greater the bandwidth transmitted, the greater the cost. Thus, on the one hand, economic and practical reasons dictate that digital information be approximated by a signal of limited bandwidth. On the other hand, limiting the bandwidth creates distortions, which makes the task of interpreting the received signal more difficult. The more limited the bandwidth, the greater the distortion, and the greater the potential for error by the receiver.

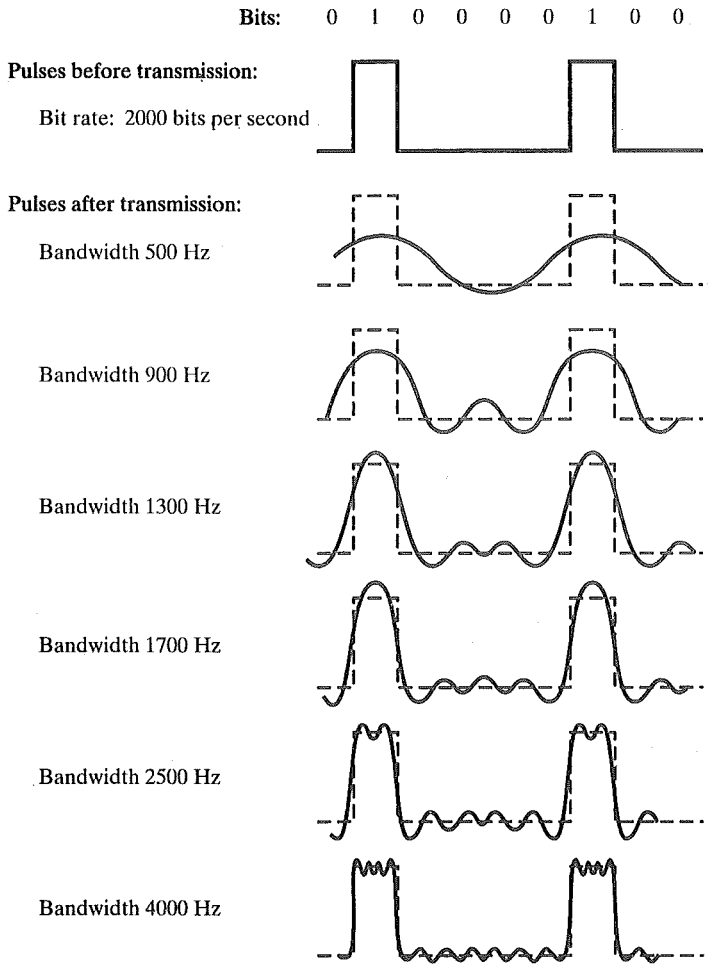


FIGURE 2.9 Effect of bandwidth on a digital signal.

One more illustration should serve to reinforce these concepts. Figure 2.9 shows a digital bit stream with a data rate of 2000 bits per second. With a bandwidth of 1700 to 2500 Hz, the representation is quite good. Furthermore, we can generalize these results. If the data rate of the digital signal is W bps, then a very good representation can be achieved with a bandwidth of $2W$ Hz; however, unless noise is very severe, the bit pattern can be recovered with less bandwidth than this.

Thus, there is a direct relationship between data rate and bandwidth: the higher the data rate of a signal, the greater is its effective bandwidth. Looked at the other way, the greater the bandwidth of a transmission system, the higher is the data rate that can be transmitted over that system.

Another observation worth making is this: If we think of the bandwidth of a signal as being centered about some frequency, referred to as the *center frequency*, then the higher the center frequency, the higher the potential bandwidth and therefore the higher the potential data rate. Consider that if a signal is centered at 2 MHz, its maximum bandwidth is 4 MHz.

We return to a discussion of the relationship between bandwidth and data rate later in this chapter, after a consideration of transmission impairments.

2.2 ANALOG AND DIGITAL DATA TRANSMISSION

In transmitting data from a source to a destination, one must be concerned with the nature of the data, the actual physical means used to propagate the data, and what processing or adjustments may be required along the way to assure that the received data are intelligible. For all of these considerations, the crucial question is whether we are dealing with analog or digital entities.

The terms *analog* and *digital* correspond, roughly, to *continuous* and *discrete*, respectively. These two terms are used frequently in data communications in at least three contexts:

- Data
- Signaling
- Transmission

We discussed data, as distinct from information, in Chapter 1. For present purposes, we define data as entities that convey meaning. Signals are electric or electromagnetic encoding of data. Signaling is the act of propagating the signal along a suitable medium. Finally, transmission is the communication of data by the propagation and processing of signals. In what follows, we try to make these abstract concepts clear by discussing the terms *analog* and *digital* in these three contexts.

Data

The concepts of analog and digital data are simple enough. Analog data take on continuous values on some interval. For example, voice and video are continuously varying patterns of intensity. Most data collected by sensors, such as temperature and pressure, are continuous-valued. Digital data take on discrete values; examples are text and integers.

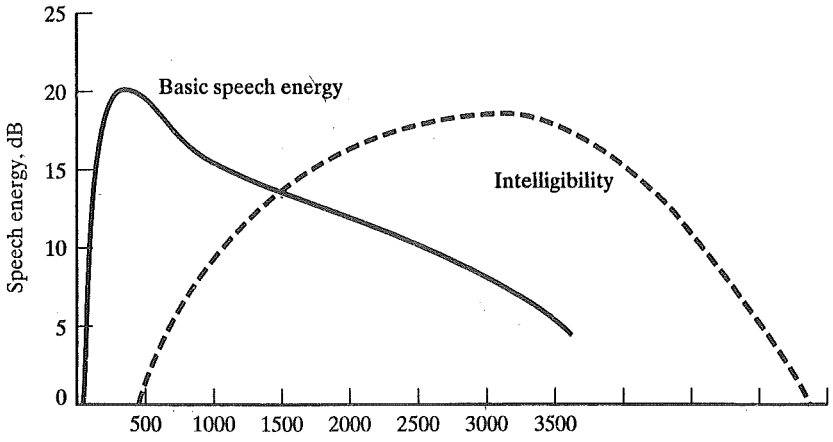


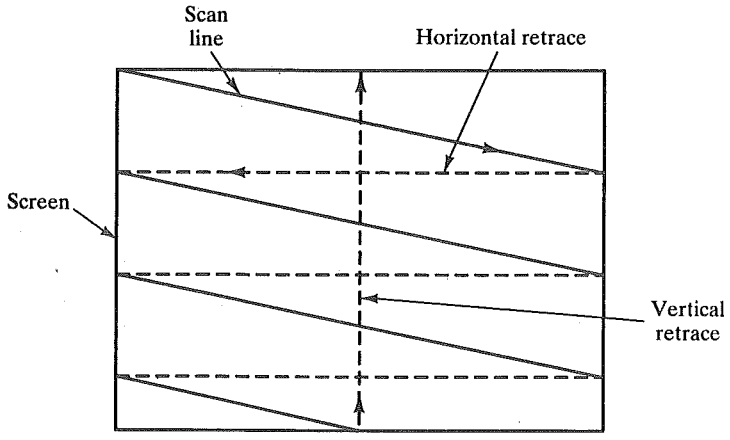
FIGURE 2.10 Acoustic spectrum for speech. *SOURCE: [FREE89]*

The most familiar example of analog data is audio or acoustic data, which, in the form of sound waves, can be perceived directly by human beings. Figure 2.10 shows the acoustic spectrum for human speech. Frequency components of speech may be found between 20 Hz and 20 kHz. Although much of the energy in speech is concentrated at the lower frequencies, tests have shown that frequencies up to 600 to 700 Hz add very little to the intelligibility of speech to the human ear. The dashed line more accurately reflects the intelligibility or emotional content of speech.

Another common example of analog data is video. Here it is easier to characterize the data in terms of the viewer (destination) of the TV screen rather than the original scene (source) that is recorded by the TV camera. To produce a picture on the screen, an electron beam scans across the surface of the screen from left to right and top to bottom. For black-and-white television, the amount of illumination produced (on a scale from black to white) at any point is proportional to the intensity of the beam as it passes that point. Thus, at any instant in time, the beam takes on an analog value of intensity to produce the desired brightness at that point on the screen. Further, as the beam scans, the analog value changes. The video image, then, can be viewed as a time-varying analog signal.

Figure 2.11a depicts the scanning process. At the end of each scan line, the beam is swept rapidly back to the left (horizontal retrace). When the beam reaches the bottom, it is swept rapidly back to the top (vertical retrace). The beam is turned off (blanked out) during the retrace intervals.

To achieve adequate resolution, the beam produces a total of 483 horizontal lines at a rate of 30 complete scans of the screen per second. Tests have shown that this rate will produce a sensation of flicker rather than smooth motion. However, the flicker is eliminated by a process of interlacing, as depicted in Figure 2.11b. The electron beam scans across the screen starting at the far left, very near the top. The beam reaches the bottom at the middle after 241½ lines. At this point, the beam is quickly repositioned at the top of the screen and, beginning in the middle, produces an additional 241½ lines interlaced with the original set. Thus, the screen is



(a) Composition of a TV field

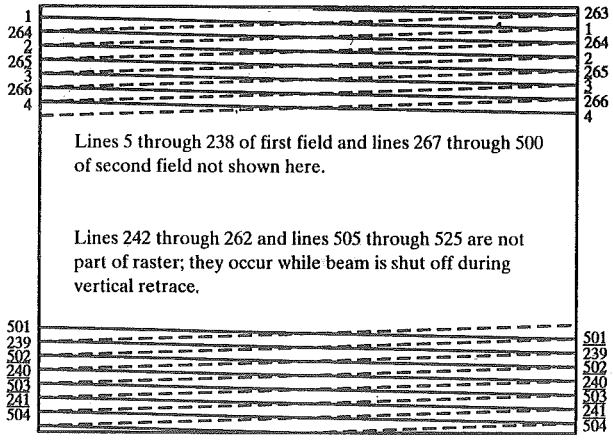


FIGURE 2.11 TV picture production.

refreshed 60 times per second rather than 30, and flicker is avoided. Note that the total count of lines is 525. Of these, 42 are blanked out during the vertical retrace interval, leaving 483 actually visible on the screen.

A familiar example of digital data is text or character strings. While textual data are most convenient for human beings, they cannot, in character form, be easily stored or transmitted by data processing and communications systems. Such systems are designed for binary data. Thus, a number of codes have been devised by which characters are represented by a sequence of bits. Perhaps the earliest common example of this is the Morse code. Today, the most commonly used code in the United States is the ASCII (American Standard Code for Information Interchange) (Table 2.1) promulgated by ANSI. ASCII is also widely used outside the United States. Each character in this code is represented by a unique 7-bit pattern; thus, 128 different characters can be represented. This is a larger number than is necessary, and some of the patterns represent "control" characters (Table 2.2). Some of these

TABLE 2.1 The American Standard Code for Information Interchange (ASCII).

bit position															
						0	0	0	0	1	1	1	1
						0	0	1	1	0	0	1	1
						0	1	0	1	0	1	0	1
b_7	b_6	b_5	b_4	b_3	b_2	b_1							
			0	0	0	0	NUL	DLE	SP	0	@	P	‘	p
			0	0	0	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	STX	DC2	“	2	B	R	b	r
			0	0	1	1	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	BEL	ETB	‘	7	G	W	g	w
			1	0	0	0	BS	CAN	(	8	H	X	h	x
			1	0	0	1	HT	EM	)	9	I	Y	i	y
			1	0	1	0	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	VT	ESC	+	;	K	[	k	{
			1	1	0	0	FF	FS	,	<	L	\	l	|
			1	1	0	1	CR	GS	–	=	M	]	m	}
			1	1	1	0	SO	RS	.	>	N	^	n	~
			1	1	1	1	SI	US	/	?	O	_	o	DEL

This is the U.S. national version of CCITT International Alphabet Number 5 (T.50). Control characters are explained in Table 2.

control characters have to do with controlling the printing of characters on a page. Others are concerned with communications procedures and will be discussed later. ASCII-encoded characters are almost always stored and transmitted using 8 bits per character (a block of 8 bits is referred to as an octet or a byte). The eighth bit is a parity bit used for error detection. This bit is set such that the total number of binary 1s in each octet is always odd (odd parity) or always even (even parity). Thus, a transmission error that changes a single bit can be detected.

Signals

In a communications system, data are propagated from one point to another by means of electric signals. An analog signal is a continuously varying electromagnetic wave that may be propagated over a variety of media, depending on spectrum; examples are wire media, such as twisted pair and coaxial cable, fiber optic cable,

TABLE 2.2 ASCII control characters. (*Continued on next page.*)

Format control

BS (Backspace): Indicates movement of the printing mechanism or display cursor backward one position.

HT (Horizontal Tab): Indicates movement of the printing mechanism or display cursor forward to the next preassigned 'tab' or stopping position.

LF (Line Feed): Indicates movement of the printing mechanism or display cursor to the start of the next line.

VT (Vertical Tab): Indicates movement of the printing mechanism or display cursor to the next of a series preassigned printing lines.

FF (Form Feed): Indicates movement of the printing mechanism or display cursor to the starting position of the next page, form, or screen.

CR (Carriage Return): Indicates movement of the printing mechanism or display cursor to the starting position of the same line.

Transmission control

SOH (Start of Heading): Used to indicate the start of a heading, which may contain address or routing information.

STX (Start of Text): Used to indicate the start of the text and so also indicates the end of the heading.

ETX (End of Text): Used to terminate the text that was started with STX.

EOT (End of Transmission): Indicates the end of a transmission, which may have included one or more 'texts' with their headings.

ENQ (Enquiry): A request for a response from a remote station. It may be used as a 'WHO ARE YOU' request for a station to identify itself.

ACK (Acknowledge): A character transmitted by a receiving device as an affirmation response to a sender. It is used as a positive response to polling messages.

NAK (Negative Acknowledgment): A character transmitted by a receiving device as a negative response to a sender. It is used as a negative response to polling messages.

SYN (Synchronous/Idle): Used by a synchronous transmission system to achieve synchronization. When no data are being sent, a synchronous transmission system may send SYN characters continuously.

ETB (End of Transmission Block): Indicates the end of a block of data for communication purposes. It is used for blocking data where the block structure is not necessarily related to the processing format.

Information separator

FS (File Separator)
GS (Group Separator)
RS (Record Separator)
US (United Separator)

Information separators to be used in an optional manner except that their hierarchy shall be FS (the most inclusive) to US (the least inclusive).

and atmosphere or space propagation. A digital signal is a sequence of voltage pulses that may be transmitted over a wire medium; for example, a constant positive voltage level may represent binary 1, and a constant negative voltage level may represent binary 0.

In what follows, we look first at some specific examples of signal types and then discuss the relationship between data and signals.

Examples

Let us return to our three examples of the preceding subsection. For each example, we will describe the signal and estimate its bandwidth.

In the case of acoustic data (voice), the data can be represented directly by an electromagnetic signal occupying the same spectrum. Although, there is a need to

TABLE 2.2 (*Continued*).

Miscellaneous

NUL (Null): No character. Used for filling in time or filling space on tape when there are no data.

BEL (Bell): Used when there is need to call human attention. It may control alarm or attention devices.

SO (Shift Out): Indicates that the code combinations that follow shall be interpreted as outside of the standard character set until an SI character is reached.

SI (Shift In): Indicates that the code combinations that follow shall be interpreted according to the standard character set.

DEL (Delete): Used to obliterate unwanted characters, for example, by overwriting.

SP (Space): A nonprinting character used to separate words, or to move the printing mechanism or display cursor forward by one position.

DLE (Data Link Escape): A character that shall change the meaning of one or more contiguously following characters. It can provide supplementary controls or permit the sending of data characters having any bit combination.

DC1, DC2, DC3, DC4 (Device Controls): Characters for the control of ancillary devices or special terminal features.

CAN (Cancel): Indicates that the data that precede it in a message or block should be disregarded (usually because an error has been detected).

EM (End of Medium): Indicates the physical end of a tape or other medium, or the end of the required or used portion of the medium.

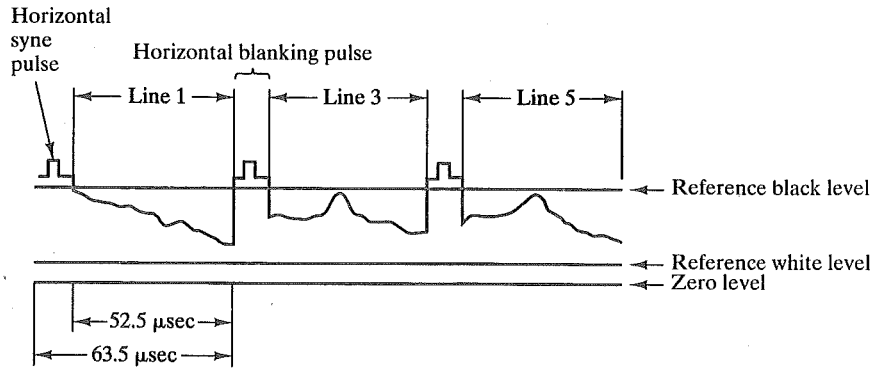
SUB (Substitute): Substituted for a character that is found to be erroneous or invalid.

ESC (Escape): A character intended to provide code extension in that it gives a specified number of continuously following characters an alternate meaning.

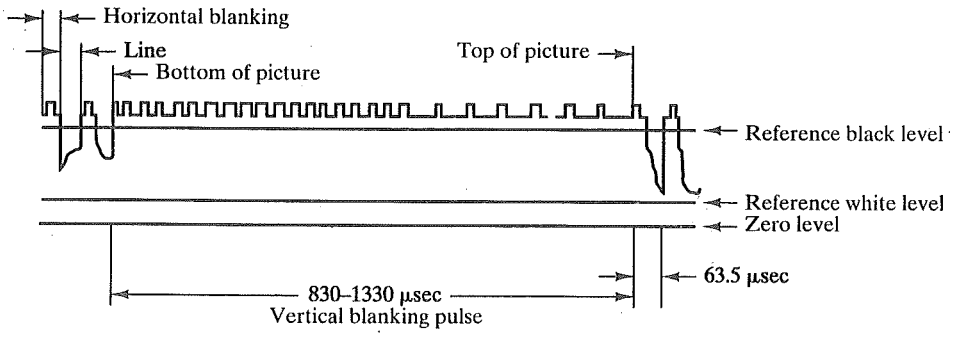
compromise between the fidelity of the sound, as transmitted electrically, and the cost of transmission, which increases with increasing bandwidth. Although, as mentioned, the spectrum of speech is approximately 20 Hz to 20 kHz, a much narrower bandwidth will produce acceptable voice reproduction. The standard spectrum for a voice signal is 300 to 3400 Hz. This is adequate for voice reproduction, it minimizes required transmission capacity, and it allows for the use of rather inexpensive telephone sets. Thus, the telephone transmitter converts the incoming acoustic voice signal into an electromagnetic signal over the range 300 to 3400 Hz. This signal is then transmitted through the telephone system to a receiver, which reproduces an acoustic signal from the incoming electromagnetic signal.

Now, let us look at the video signal, which, interestingly, consists of both analog and digital components. To produce a video signal, a TV camera, which performs similar functions to the TV receiver, is used. One component of the camera is a photosensitive plate, upon which a scene is optically focused. An electron beam sweeps across the plate from left to right and top to bottom, in the same fashion as depicted in Figure 2.11 for the receiver. As the beam sweeps, an analog electric signal is developed proportional to the brightness of the scene at a particular spot.

Now we are in a position to describe the video signal. Figure 2.12a shows three lines of a video signal; in this diagram, white is represented by a small positive voltage, and black by a much larger positive voltage. So, for example, line 3 is at a medium gray level most of the way across with a blacker portion in the middle. Once the beam has completed a scan from left to right, it must retrace to the left edge to scan the next line. During this period, the picture should be blanked out (on both camera and receiver). This is done with a digital "horizontal blanking pulse."



(a) Horizontal lines of video



(b) Vertical blanking signal

FIGURE 2.12 Video signal (different scales for a and b).

Also, to maintain transmitter-receiver synchronization, a synchronization (sync) pulse is sent between every line of video signal. This horizontal sync pulse rides on top of the blanking pulse, creating a staircase-shaped digital signal between adjacent analog video signals. Finally, when the beam reaches the bottom of the screen, it must return to the top, with a somewhat longer blanking interval required. This is shown in Figure 2.12b. The vertical blanking pulse is actually a series of synchronization and blanking pulses, whose details need not concern us here.

Next, consider the timing of the system. We mentioned that a total of 483 lines are scanned at a rate of 30 complete scans per second. This is an approximate number taking into account the time lost during the vertical retrace interval. The actual U.S. standard is 525 lines, but of these about 42 are lost during vertical retrace. Thus, the horizontal scanning frequency is $\frac{525 \text{ lines}}{1/30 \text{ s/scan}} = 15{,}750$ lines per second, or 63.5 μs/line. Of this 63.5 μs, about 11 μs are allowed for horizontal retrace, leaving a total of 52.5 μs per video line.

Finally, we are in a position to estimate the bandwidth required for the video signal. To do this, we must estimate the upper (maximum) and lower (minimum)

frequency of the band. We use the following reasoning to arrive at the maximum frequency: The maximum frequency would occur during the horizontal scan if the scene were alternating between black and white as rapidly as possible. We can estimate this maximum value by considering the resolution of the video image. In the vertical dimension, there are 483 lines, so the maximum vertical resolution would be 483. Experiments have shown that the actual subjective resolution is about 70 percent of that number, or about 338 lines. In the interest of a balanced picture, the horizontal and vertical resolutions should be about the same. Because the ratio of width to height of a TV screen is 4:3, the horizontal resolution should be about $4/3 \times 338 = 450$ lines. As a worst case, a scanning line would be made up of 450 elements alternating black and white. The scan would result in a wave, with each cycle of the wave consisting of one higher (black) and one lower (white) voltage level. Thus, there would be $450/2 = 225$ cycles of the wave in 52.5 μs, for a maximum frequency of about 4 MHz. This rough reasoning, in fact, is fairly accurate. The maximum frequency, then, is 4 MHz. The lower limit will be a dc or zero frequency, where the dc component corresponds to the average illumination of the scene (the average value by which the signal exceeds the reference white level). Thus, the bandwidth of the video signal is approximately 4 MHz − 0 = 4 MHz.

The foregoing discussion did not consider color or audio components of the signal. It turns out that, with these included, the bandwidth remains about 4 MHz.

Finally, the third example described above is the general case of binary digital data. A commonly used signal for such data uses two constant (dc) voltage levels, one level for binary 1 and one level for binary 0. (In Chapter 3, we shall see that this is but one alternative, referred to as NRZ.) Again, we are interested in the bandwidth of such a signal. This will depend, in any specific case, on the exact shape of the waveform and on the sequence of 1s and 0s. We can obtain some understanding by considering Figure 2.9 (compare Figure 2.8). As can be seen, the greater the bandwidth of the signal, the more faithfully it approximates a digital pulse stream.

Data and Signals

In the foregoing discussion, we have looked at analog signals used to represent analog data and digital signals used to represent digital data. Generally, analog data are a function of time and occupy a limited frequency spectrum; such data can be represented by an electromagnetic signal occupying the same spectrum. Digital data can be represented by digital signals, with a different voltage level for each of the two binary digits.

As Figure 2.13 illustrates, these are not the only possibilities. Digital data can also be represented by analog signals by use of a modem (modulator/demodulator). The modem converts a series of binary (two-valued) voltage pulses into an analog signal by encoding the digital data onto a carrier frequency. The resulting signal occupies a certain spectrum of frequency centered about the carrier and may be propagated across a medium suitable for that carrier. The most common modems represent digital data in the voice spectrum and, hence, allow those data to be propagated over ordinary voice-grade telephone lines. At the other end of the line, the modem demodulates the signal to recover the original data.

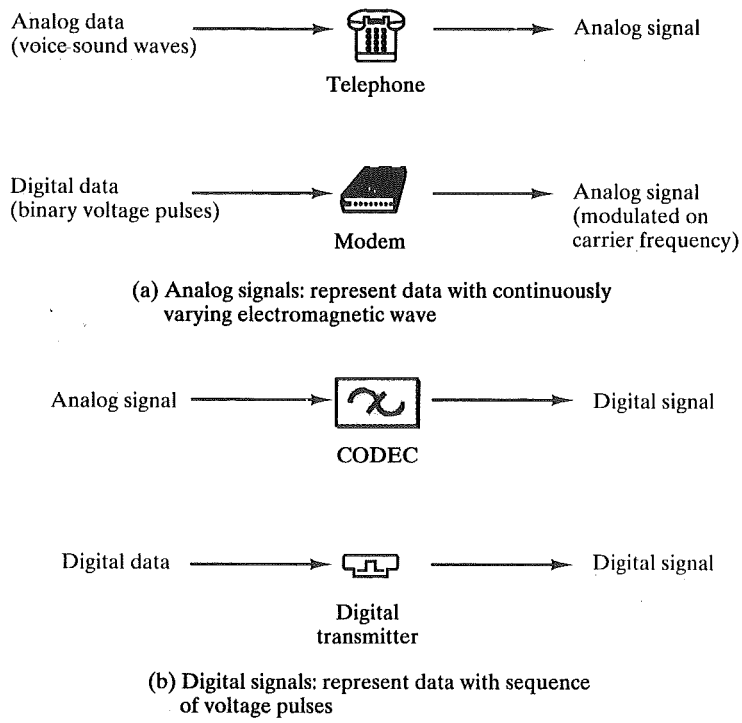


FIGURE 2.13 Analog and digital signaling of analog and digital data.

In an operation very similar to that performed by a modem, analog data can be represented by digital signals. The device that performs this function for voice data is a codec (coder-decoder). In essence, the codec takes an analog signal that directly represents the voice data and approximates that signal by a bit stream. At the receiving end, the bit stream is used to reconstruct the analog data.

Thus, Figure 2.13 suggests that data may be encoded into signals in a variety of ways. We will return to this topic in Chapter 4.

Transmission

A final distinction remains to be made. Both analog and digital signals may be transmitted on suitable transmission media. The way these signals are treated is a function of the transmission system. Table 2.3 summarizes the methods of data transmission. Analog transmission is a means of transmitting analog signals without regard to their content; the signals may represent analog data (e.g., voice) or digital data (e.g., binary data that pass through a modem). In either case, the analog signal will become weaker (attenuated) after a certain distance. To achieve longer distances, the analog transmission system includes amplifiers that boost the energy in the signal. Unfortunately, the amplifier also boosts the noise components. With amplifiers cascaded to achieve long distances, the signal becomes more and more distorted. For analog data, such as voice, quite a bit of distortion can be tolerated

TABLE 2.3 Analog and digital transmission.

	Analog signal	Digital signal
Analog data	Two alternatives: (1) signal occupies the same spectrum as the analog data; (2) analog data are encoded to occupy a different portion of spectrum.	Analog data are encoded using a codec to produce a digital bit stream.
Digital data	Digital data are encoded using a modem to produce analog signal.	Two alternatives: (1) signal consists of two voltage levels to represent the two binary values; (2) digital data are encoded to produce a digital signal with desired properties.

(a) Data and signals

	Analog transmission	Digital transmission
Analog signal	Is propagated through amplifiers; same treatment whether signal is used to represent analog data or digital data.	Assumes that the analog signal represents digital data. Signal is propagated through repeaters; at each repeater, digital data are recovered from inbound signal and used to generate a new analog outbound signal.
Digital signal	Not used	Digital signal represents a stream of 1s and 0s, which may represent digital data or may be an encoding of analog data. Signal is propagated through repeaters; at each repeater, stream of 1s and 0s is recovered from inbound signal and used to generate a new digital outbound signal.

(b) Treatment of signals

and the data remain intelligible. However, for digital data, cascaded amplifiers will introduce errors.

Digital transmission, in contrast, is concerned with the content of the signal. A digital signal can be transmitted only a limited distance before attenuation endangers the integrity of the data. To achieve greater distances, repeaters are used. A repeater receives the digital signal, recovers the pattern of 1s and 0s, and retransmits a new signal, thereby overcoming the attenuation.

The same technique may be used with an analog signal if it is assumed that the signal carries digital data. At appropriately spaced points, the transmission system has repeaters rather than amplifiers. The repeater recovers the digital data from the analog signal and generates a new, clean analog signal. Thus, noise is not cumulative.

The question naturally arises as to which is the preferred method of transmission; the answer being supplied by the telecommunications industry and its customers is digital, this despite an enormous investment in analog communications facilities. Both long-haul telecommunications facilities and intrabuilding services are gradually being converted to digital transmission and, where possible, digital signaling techniques. The most important reasons are

- **Digital technology.** The advent of large-scale integration (LSI) and very large-scale integration (VLSI) technology has caused a continuing drop in the cost and size of digital circuitry. Analog equipment has not shown a similar drop.
- **Data integrity.** With the use of repeaters rather than amplifiers, the effects of noise and other signal impairments are not cumulative. It is possible, then, to transmit data longer distances and over lesser quality lines by digital means while maintaining the integrity of the data. This is explored in Section 2.3.
- **Capacity utilization.** It has become economical to build transmission links of very high bandwidth, including satellite channels and connections involving optical fiber. A high degree of multiplexing is needed to effectively utilize such capacity, and this is more easily and cheaply achieved with digital (time-division) rather than analog (frequency-division) techniques. This is explored in Chapter 7.
- **Security and privacy.** Encryption techniques can be readily applied to digital data and to analog data that have been digitized.
- **Integration.** By treating both analog and digital data digitally, all signals have the same form and can be treated similarly. Thus, economies of scale and convenience can be achieved by integrating voice, video, and digital data.

2.3 TRANSMISSION IMPAIRMENTS

With any communications system, it must be recognized that the received signal will differ from the transmitted signal due to various transmission impairments. For analog signals, these impairments introduce various random modifications that degrade the signal quality. For digital signals, bit errors are introduced: A binary 1 is transformed into a binary 0 and vice versa. In this section, we examine the various impairments and comment on their effect on the information-carrying capacity of a communication link; the next chapter looks at measures to compensate for these impairments.

The most significant impairments are

- Attenuation and attenuation distortion
- Delay distortion
- Noise

Attenuation

The strength of a signal falls off with distance over any transmission medium. For guided media, this reduction in strength, or attenuation, is generally logarithmic and is thus typically expressed as a constant number of decibels per unit distance. For unguided media, attenuation is a more complex function of distance and of the makeup of the atmosphere. Attenuation introduces three considerations for the transmission engineer. First, a received signal must have sufficient strength so that the electronic circuitry in the receiver can detect and interpret the signal. Second, the signal must maintain a level sufficiently higher than noise to be received without error. Third, attenuation is an increasing function of frequency.

The first and second problems are dealt with by attention to signal strength and by the use of amplifiers or repeaters. For a point-to-point link, the signal strength of the transmitter must be strong enough to be received intelligibly, but not so strong as to overload the circuitry of the transmitter, which would cause a distorted signal to be generated. Beyond a certain distance, the attenuation is unacceptably great, and repeaters or amplifiers are used to boost the signal from time to time. These problems are more complex for multipoint lines where the distance from transmitter to receiver is variable.

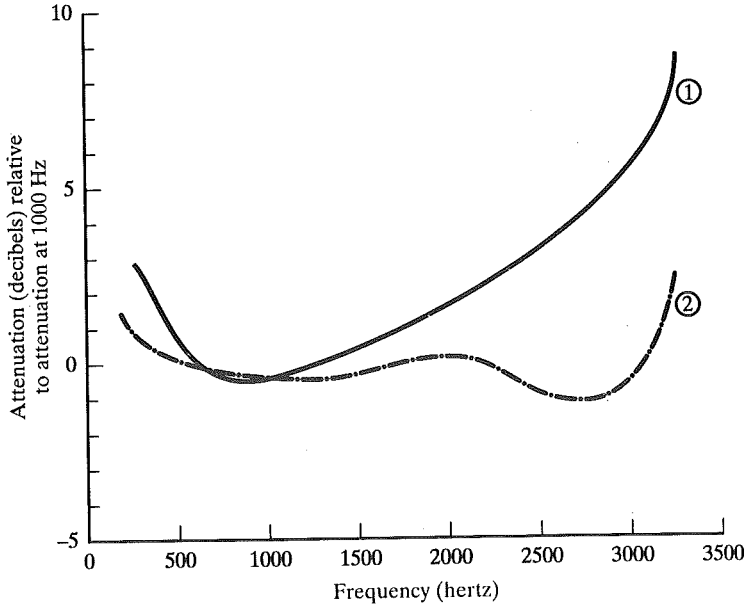
The third problem is particularly noticeable for analog signals. Because the attenuation varies as a function of frequency, the received signal is distorted, reducing intelligibility. To overcome this problem, techniques are available for equalizing attenuation across a band of frequencies. This is commonly done for voice-grade telephone lines by using loading coils that change the electrical properties of the line; the result is to smooth out attenuation effects. Another approach is to use amplifiers that amplify high frequencies more than lower frequencies.

An example is shown in Figure 2.14a, which shows attenuation as a function of frequency for a typical leased line. In the figure, attenuation is measured relative to the attenuation at 1000 Hz. Positive values on the y axis represent attenuation greater than that at 1000 Hz. A 1000-Hz tone of a given power level is applied to the input, and the power, P_{1000}, is measured at the output. For any other frequency f, the procedure is repeated and the relative attenuation in decibels is

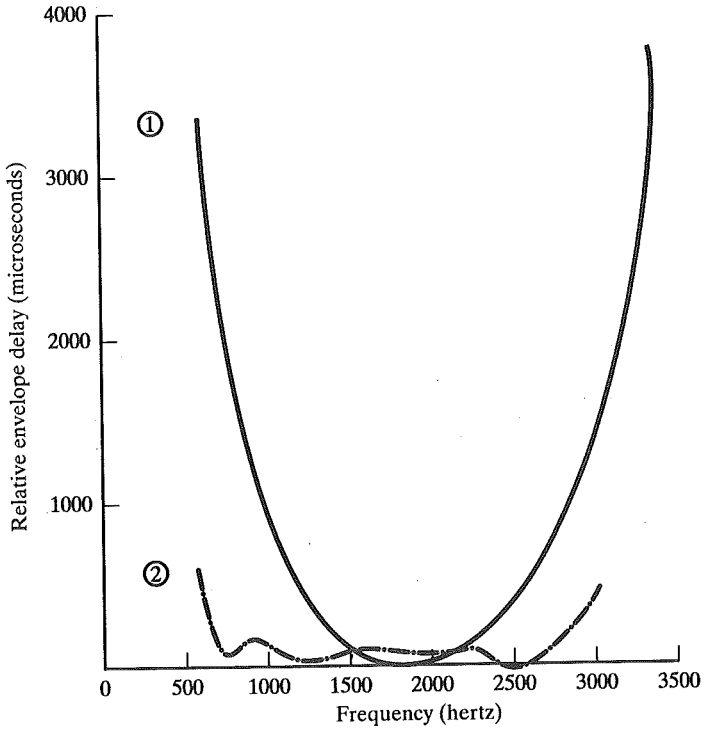
$$N_f = -10 \log_{10} \frac{P_f}{P_{1000}}$$

The solid line in Figure 2.14a shows attenuation without equalization. As can be seen, frequency components at the upper end of the voice band are attenuated much more than those at lower frequencies. It should be clear that this will result in a distortion of the received speech signal. The dashed line shows the effect of equalization. The flattened response curve improves the quality of voice signals. It also allows higher data rates to be used for digital data that are passed through a modem.

Attenuation distortion is much less of a problem with digital signals. As we have seen, the strength of a digital signal falls off rapidly with frequency (Figure 2.6b); most of the content is concentrated near the fundamental frequency, or bit rate, of the signal.



(a) Attenuation



(b) Delay distortion

FIGURE 2.14
Attenuation and delay distortion curves for a voice channel.

Delay Distortion

Delay distortion is a phenomenon peculiar to guided transmission media. The distortion is caused by the fact that the velocity of propagation of a signal through a guided medium varies with frequency. For a bandlimited signal, the velocity tends to be highest near the center frequency and lower toward the two edges of the band. Thus, various frequency components of a signal will arrive at the receiver at different times.

This effect is referred to as delay distortion, as the received signal is distorted due to variable delay in its components. Delay distortion is particularly critical for digital data. Consider that a sequence of bits is being transmitted, using either analog or digital signals. Because of delay distortion, some of the signal components of one bit position will spill over into other bit positions, causing intersymbol interference, which is a major limitation to maximum bit rate over a transmission control.

Equalizing techniques can also be used for delay distortion. Again using a leased telephone line as an example, Figure 2.14b shows the effect of equalization on delay as a function of frequency.

Noise

For any data transmission event, the received signal will consist of the transmitted signal, modified by the various distortions imposed by the transmission system, plus additional unwanted signals that are inserted somewhere between transmission and reception; the latter, undesired signals are referred to as noise—a major limiting factor in communications system performance.

Noise may be divided into four categories:

- Thermal noise
- Intermodulation noise
- Crosstalk
- Impulse noise

Thermal noise is due to thermal agitation of electrons in a conductor. It is present in all electronic devices and transmission media and is a function of temperature. Thermal noise is uniformly distributed across the frequency spectrum and hence is often referred to as white noise; it cannot be eliminated and therefore places an upper bound on communications system performance. The amount of thermal noise to be found in a bandwidth of 1 Hz in any device or conductor is

$$N_0 = kT$$

where

N_0 = noise power density, watts/hertz
k = Boltzmann's constant = 1.3803×10^{-23} J/°K
T = temperature, degrees Kelvin

The noise is assumed to be independent of frequency. Thus, the thermal noise, in watts, present in a bandwidth of W hertz can be expressed as

$$N = kTW$$

or, in decibel-watts,

$$\begin{aligned} N &= 10 \log k + 10 \log T + 10 \log W \\ &= -228.6 \text{ dBW} + 10 \log T + 10 \log W \end{aligned}$$

When signals at different frequencies share the same transmission medium, the result may be intermodulation noise. The effect of intermodulation noise is to produce signals at a frequency that is the sum or difference of the two original frequencies, or multiples of those frequencies. For example, the mixing of signals at frequencies f_1 and f_2 might produce energy at the frequency $f_1 + f_2$. This derived signal could interfere with an intended signal at the frequency $f_1 + f_2$.

Intermodulation noise is produced when there is some nonlinearity in the transmitter, receiver, or intervening transmission system. Normally, these components behave as linear systems; that is, the output is equal to the input, times a constant. In a nonlinear system, the output is a more complex function of the input. Such nonlinearity can be caused by component malfunction or the use of excessive signal strength. It is under these circumstances that the sum and difference terms occur.

Crosstalk has been experienced by anyone who, while using the telephone, has been able to hear another conversation; it is an unwanted coupling between signal paths. It can occur by electrical coupling between nearby twisted pair or, rarely, coax cable lines carrying multiple signals. Crosstalk can also occur when unwanted signals are picked up by microwave antennas; although highly directional, microwave energy does spread during propagation. Typically, crosstalk is of the same order of magnitude (or less) as thermal noise.

All of the types of noise discussed so far have reasonably predictable and reasonably constant magnitudes; it is thus possible to engineer a transmission system to cope with them. Impulse noise, however, is noncontinuous, consisting of irregular pulses or noise spikes of short duration and of relatively high amplitude. It is generated from a variety of causes, including external electromagnetic disturbances, such as lightning, and faults and flaws in the communications system.

Impulse noise is generally only a minor annoyance for analog data. For example, voice transmission may be corrupted by short clicks and crackles with no loss of intelligibility. However, impulse noise is the primary source of error in digital data communication. For example, a sharp spike of energy of 0.01-second duration would not destroy any voice data, but would wash out about 50 bits of data being transmitted at 4800 bps. Figure 2.15 is an example of the effect on a digital signal. Here the noise consists of a relatively modest level of thermal noise plus occasional spikes of impulse noise. The digital data are recovered from the signal by sampling

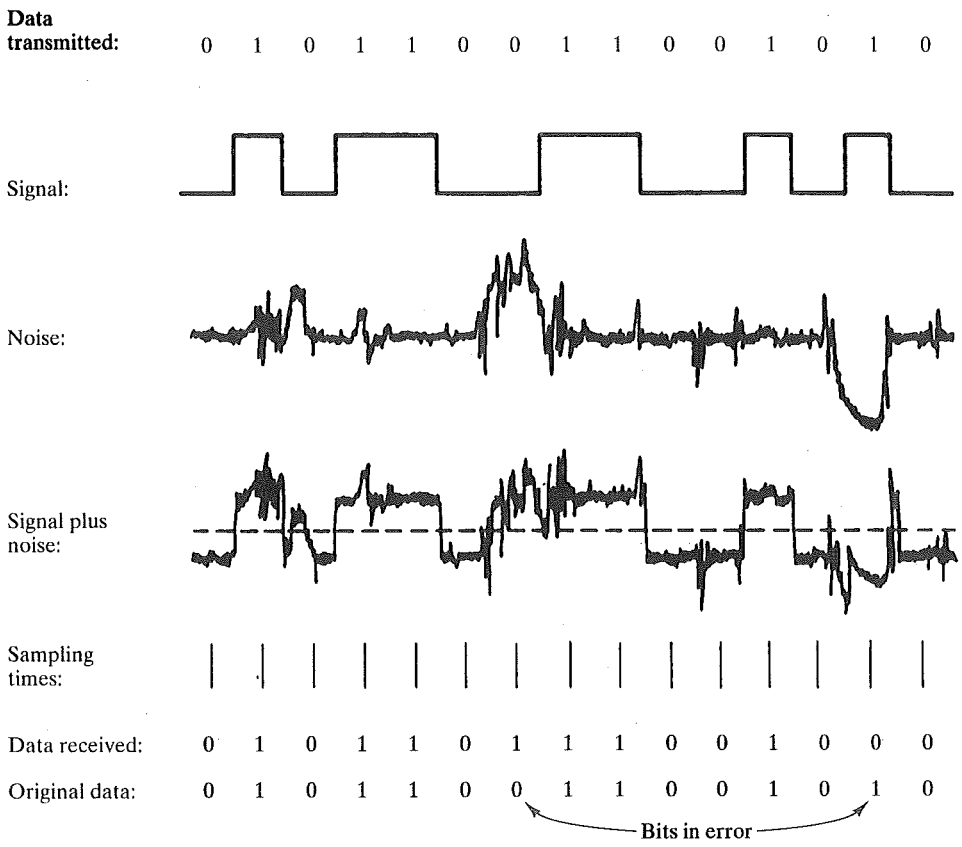


FIGURE 2.15 Effect of noise on a digital signal.

the received waveform once per bit time. As can be seen, the noise is occasionally sufficient to change a 1 to a 0 or a 0 to a 1.

Channel Capacity

We have seen that there are a variety of impairments that distort or corrupt a signal. For digital data, the question that then arises is to what extent these impairments limit the data rate that can be achieved. The rate at which data can be transmitted over a given communication path, or channel, under given conditions, is referred to as the channel capacity.

There are four concepts here that we are trying to relate to one another:

- **Data rate.** This is the rate, in bits per second (bps), at which data can be communicated.
- **Bandwidth.** This is the bandwidth of the transmitted signal as constrained by the transmitter and by the nature of the transmission medium, expressed in cycles per second, or hertz.

- **Noise.** The average level of noise over the communications path.
- **Error rate.** The rate at which errors occur, where an error is the reception of a 1 when a 0 was transmitted, or the reception of a 0 when a 1 was transmitted.

The problem we are addressing is this: Communications facilities are expensive, and, in general, the greater the bandwidth of a facility, the greater the cost. Furthermore, all transmission channels of any practical interest are of limited bandwidth. The limitations arise from the physical properties of the transmission medium or from deliberate limitations at the transmitter on the bandwidth to prevent interference from other sources. Accordingly, we would like to make as efficient use as possible of a given bandwidth. For digital data, this means that we would like to get as high a data rate as possible at a particular limit of error rate for a given bandwidth. The main constraint on achieving this efficiency is noise.

To begin, let us consider the case of a channel that is noise-free. In this environment, the limitation on data rate is simply the bandwidth of the signal. A formulation of this limitation, due to Nyquist, states that if the rate of signal transmission is $2W$, then a signal with frequencies no greater than W is sufficient to carry the data rate. The converse is also true: Given a bandwidth of W, the highest signal rate that can be carried is $2W$. This limitation is due to the effect of intersymbol interference, such as is produced by delay distortion. The result is useful in the development of digital-to-analog encoding schemes and is derived in Appendix 4A.

Note that in the last paragraph, we referred to signal rate. If the signals to be transmitted are binary (two voltage levels), then the data rate that can be supported by W Hz is $2W$ bps. As an example, consider a voice channel being used, via modem, to transmit digital data. Assume a bandwidth of 3100 Hz. Then the capacity, C, of the channel is $2W = 6200$ bps. However, as we shall see in Chapter 4, signals with more than two levels can be used; that is, each signal element can represent more than one bit. For example, if four possible voltage levels are used as signals, then each signal element can represent two bits. With multilevel signaling, the Nyquist formulation becomes

$$C = 2W \log_2 M$$

where M is the number of discrete signal or voltage levels. Thus, for $M = 8$, a value used with some modems, C becomes 18,600 bps.

So, for a given bandwidth, the data rate can be increased by increasing the number of different signals. However, this places an increased burden on the receiver: Instead of distinguishing one of two possible signals during each signal time, it must distinguish one of M possible signals. Noise and other impairments on the transmission line will limit the practical value of M.

Thus, all other things being equal, doubling the bandwidth doubles the data rate. Now consider the relationship between data rate, noise, and error rate. This can be explained intuitively by again considering Figure 2.15. The presence of noise can corrupt one or more bits. If the data rate is increased, then the bits become "shorter" so that more bits are affected by a given pattern of noise. Thus, at a given noise level, the higher the data rate, the higher the error rate.

All of these concepts can be tied together neatly in a formula developed by the mathematician Claude Shannon. As we have just illustrated, the higher the data rate, the more damage that unwanted noise can do. For a given level of noise, we would expect that a greater signal strength would improve the ability to correctly receive data in the presence of noise. The key parameter involved in this reasoning is the signal-to-noise ratio (S/N), which is the ratio of the power in a signal to the power contained in the noise that is present at a particular point in the transmission. Typically, this ratio is measured at a receiver, as it is at this point that an attempt is made to process the signal and eliminate the unwanted noise. For convenience, this ratio is often reported in decibels:

$$(S/N)_{dB} = 10 \log \frac{\text{signal power}}{\text{noise power}}$$

This expresses the amount, in decibels, that the intended signal exceeds the noise level. A high S/N will mean a high-quality signal and a low number of required intermediate repeaters.

The signal-to-noise ratio is important in the transmission of digital data because it sets the upper bound on the achievable data rate. Shannon's result is that the maximum channel capacity, in bits per second, obeys the equation

$$C = W \log_2\left(1 + \frac{S}{N}\right)$$

where C is the capacity of the channel in bits per second and W is the bandwidth of the channel in hertz. As an example, consider a voice channel being used, via modem, to transmit digital data. Assume a bandwidth of 3100 Hz. A typical value of S/N for a voice-grade line is 30 dB, or a ratio of 1000:1. Thus,

$$C = 3100 \log_2(1 + 1000)$$
$$= 30{,}894 \text{ bps}$$

This represents the theoretical maximum that can be achieved. In practice,however, only much lower rates are achieved. One reason for this is that the formula assumes white noise (thermal noise). Impulse noise is not accounted for, nor are attenuation or delay distortion.

The capacity indicated in the preceding equation is referred to as the error-free capacity. Shannon proved that if the actual information rate on a channel is less than the error-free capacity, then it is theoretically possible to use a suitable signal code to achieve error-free transmission through the channel. Shannon's theorem unfortunately does not suggest a means for finding such codes, but it does provide a yardstick by which the performance of practical communication schemes may be measured.

The measure of efficiency of a digital transmission is the ratio of C/W, which is the bps per hertz that is achieved. Figure 2.16 illustrates the theoretical efficiency

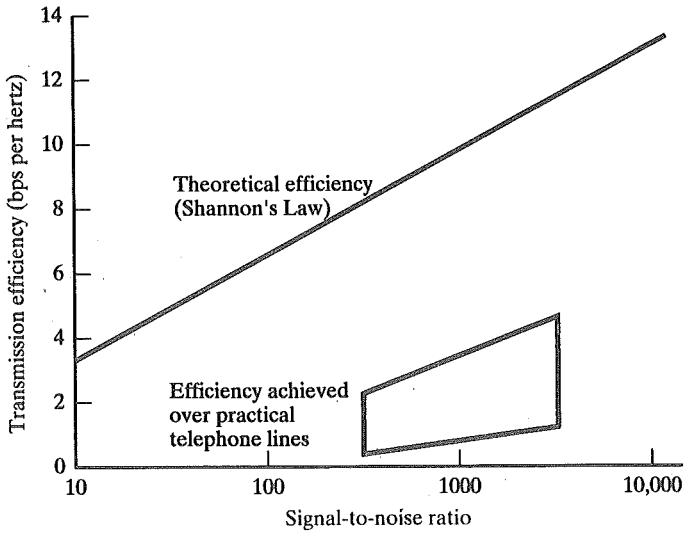


FIGURE 2.16 Theoretical and actual transmission efficiency.

of a transmission. It also shows the actual results obtained on a typical voice-grade line.

Several other observations concerning the above equation may be instructive. For a given level of noise, it would appear that the data rate could be increased by increasing either signal strength or bandwidth. However, as the signal strength increases, so do nonlinearities in the system, leading to an increase in intermodulation noise. Note also that, because noise is assumed to be white, the wider the bandwidth, the more noise is admitted to the system. Thus, as W increases, S/N decreases.

Finally, we mention a parameter related to S/N that is more convenient for determining digital data rates and error rates. The parameter is the ratio of signal energy per bit to noise-power density per hertz, E_b/N_0. Consider a signal, digital or analog, that contains binary digital data transmitted at a certain bit rate R. Recalling that 1 W = 1 J/s, the energy per bit in a signal is given by $E_b = ST_b$, where S is the signal power and T_b is the time required to send one bit. The data rate R is just $R = 1/T_b$. Thus,

$$\frac{E_b}{N_0} = \frac{S/R}{N_0} = \frac{S}{kTR}$$

or, in decibel notation,

$$\frac{E_b}{N_0} = S - 10 \log R + 228.6 \text{ dBW} - 10 \log T$$

The ratio E_b/N_0 is important because the bit error rate for digital data is a (decreasing) function of this ratio. Given a value of E_b/N_0 needed to achieve a desired error rate, the parameters in the preceding formula may be selected. Note that as the bit rate R increases, the transmitted signal power, relative to noise, must increase to maintain the required E_b/N_0.

Let us try to grasp this result intuitively by considering again Figure 2.15. The signal here is digital, but the reasoning would be the same for an analog signal. In several instances, the noise is sufficient to alter the value of a bit. Now, if the data rate were doubled, the bits would be more tightly packed together, and the same passage of noise might destroy two bits. Thus, for constant signal and noise strength, an increase in data rate increases the error rate.

Example

For binary phase-shift keying (defined in Chapter 4), $E_b/N_0 = 8.4$ dB is required for a bit error rate of 10^{-4} (probability of error = 10^{-4}). If the effective noise temperature is 290°K (room temperature) and the data rate is 2400 bps, what received signal level is required?

We have

$$\begin{aligned} 8.4 &= S(\text{dBW}) - 10 \log 2400 + 228.6 \text{ dBW} - 10 \log 290 \\ &= S(\text{dBW}) - (10)(3.38) + 228.6 - (10)(2.46) \\ S &= -161.8 \text{ dBW} \end{aligned}$$

2.4 RECOMMENDED READING

There are many books that cover the fundamentals of analog and digital transmission. [COUC95] is quite thorough. Other excellent treatments include the three-volume [BELL90], [PROA94], and [HAYK94].

BELL90 Bellcore (Bell Communications Research). *Telecommunications Transmission Engineering, Third Edition.* Three volumes. 1990.

COUC95 Couch, L. *Modern Communication Systems: Principles and Applications.* Englewood Cliffs, NJ: Prentice Hall, 1994.

HAYK94 Haykin, S. *Communication Systems.* New York: Wiley, 1994.

PROA94 Proakis, J. and Salehi, M. *Communication Systems Engineering.* Englewood Cliffs, NJ: Prentice Hall, 1994.

2.5 PROBLEMS

2.1 **a.** For the multipoint configuration of Figure 2.1, only one device at a time can transmit. Why?

b. There are two methods of enforcing the rule that only one device can transmit. In the centralized method, one station is in control and can either transmit or allow a

specified other station to transmit. In the decentralized method, the stations jointly cooperate in taking turns. What do you see as the advantages and disadvantages of the two methods?

2.2 Figure 2.6b shows the frequency-domain function for a single square pulse. The single pulse could represent a digital 1 in a communication system. Note that an infinite number of higher frequencies of decreasing magnitudes are needed to represent the single pulse. What implication does that have for a real digital transmission system?

2.3 Suppose that data are stored on 800-kbyte floppy diskettes that weigh 1 ounce each. Suppose that a Boeing 747 carries 10 tons of these floppies at a speed of 600 mph over a distance of 3000 miles. What is the data transmission rate in bits per second of this system?

2.4 ASCII is a 7-bit code that allows 128 characters to be defined. In the 1970s, many newspapers received stories from the wire services in a 6-bit code called TTS. This code carried upper- and lower-case characters as well as many special characters and formatting commands. The typical TTS character set allowed over 100 characters to be defined. How do you think this could be accomplished?

2.5 Figure 2.12 indicates that the vertical blanking pulse has a duration of 830 to 1330 μs. What is the total number of visible lines for each of these two figures?

2.6 For a video signal, what increase in horizontal resolution is possible if a bandwidth of 5 MHz is used? What increase in vertical resolution is possible? Treat the two questions separately; that is, the increased bandwidth is to be used to increase either horizontal or vertical resolution, but not both.

2.7 a. Suppose that a digitized TV picture is to be transmitted from a source that uses a matrix of 480 × 500 picture elements (pixels), where each pixel can take on one of 32 intensity values. Assume that 30 pictures are sent per second. (This digital source is roughly equivalent to broadcast TV standards that have been adopted.) Find the source rate R (bps).

b. Assume that the TV picture is to be transmitted over a channel with 4.5-MHz bandwidth and a 35-dB signal-to-noise ratio. Find the capacity of the channel (bps).

c. Discuss how the parameters given in part (a) could be modified to allow transmission of color TV signals without increasing the required value for R.

2.8 Figure 2.5 shows the effect of eliminating higher-harmonic components of a square wave and retaining only a few lower-harmonic components. What would the signal look like in the opposite case—that is, retaining all higher harmonics and eliminating a few lower harmonics?

2.9 What is the channel capacity for a teleprinter channel with a 300-Hz bandwidth and a signal-to-noise ratio of 3 dB?

2.10 A digital signaling system is required to operate at 9600 bps.

a. If a signal element encodes a 4-bit word, what is the minimum required bandwidth of the channel?

b. Repeat part (a) for the case of 8-bit words.

2.11 What is the thermal noise level of a channel with a bandwidth of 10 kHz carrying 1000 watts of power operating at 50° C?

2.12 Study the works of Shannon and Nyquist on channel capacity. Each places an upper limit on the bit rate of a channel, based on two different approaches. How are the two related?

2.13 Given a channel with an intended capacity of 20 Mbps. The bandwidth of the channel is 3 MHz. What signal-to-noise ratio is required in order to achieve this capacity?

2.14 The square wave of Figure 2.8c, with $T = 1$ msec, is transmitted through a low-pass filter that passes frequencies up to 8 kHz with no attenuation.

a. Find the power in the output waveform.

b. Assuming that at the filter input there is a thermal noise voltage with $N_0 =$ 0.1 μWatt/Hz, find the output signal-to-noise ratio in dB.

2.15 A periodic bandlimited signal has only three frequency components: dc, 100 Hz, and 200 Hz. In sine-cosine form,

$$x(t) = 12 + 15 \cos 200\pi t + 20 \sin 200\pi t - 5 \cos 400\pi t - 12 \sin 400\pi t$$

Express the signal in amplitude/phase form.

2.16 If an amplifier has a 30-dB gain, what voltage ratio does the gain represent?

2.17 An amplifier has an output of 20W. What is its output in dBW?

2A APPENDIX

FOURIER ANALYSIS

IN THIS APPENDIX, we provide an overview of key concepts in Fourier Analysis.

Fourier Series Representation of Periodic Signals

With the aid of a good table of integrals. it is a remarkably simple task to determine the frequency-domain nature of many signals. We begin with periodic signals. Any periodic signal can be represented as a sum of sinusoids, known as a Fourier series:

$$x(t) = \sum_{n=0}^{\infty} a_n \cos(2\pi n f_0 t) + \sum_{n=1}^{\infty} b_n \sin(2\pi n f_0 t)$$

where f_0 is the inverse of the period of the signal ($f_0 = 1/T$). The frequency f_0 is referred to as the *fundamental frequency*; multiples of f_0 are referred to as *harmonics*. Thus, a periodic signal with period T consists of the fundamental frequency $f_0 = 1/T$ plus harmonics of that frequency. If $a_0 \neq 0$, then $x(t)$ has a *dc component.*

The values of the coefficients are calculated as follows:

$$a_0 = \frac{1}{T}\int_0^T x(t)dt$$

$$a_n = \frac{2}{T}\int_0^T x(t)\cos(2\pi f_0 t)dt$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin(2\pi f_0 t)dt$$

This form of representation, known as the sine-cosine representation, is the easiest form to compute, but suffers from the fact that there are two components at each frequency. A more meaningful representation, the amplitude-phase representation, takes the form

$$x(t) = c_0 + \sum_{n=1}^{\infty} c_n \cos(2\pi f_0 t + \theta_n)$$

This relates to the earlier representation, as follows:

$$c_0 = a_0$$

$$c_n = \sqrt{a_n^2 + b_n^2}$$

$$\theta_n = -\tan^{-1}\left(\frac{b_n}{a_n}\right)$$

Examples of the Fourier series for periodic signals are shown in Figure 2.17.

Fourier Transform Representation of Aperiodic Signals

For a periodic signal, we have seen that its spectrum consists of discrete frequency components, at the fundamental frequency and at its harmonics. For an aperiodic signal, the spec-

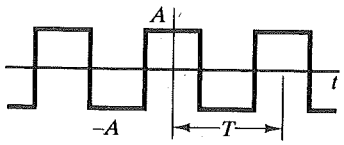
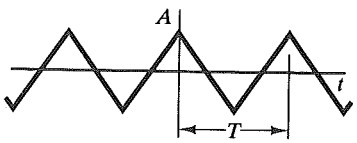
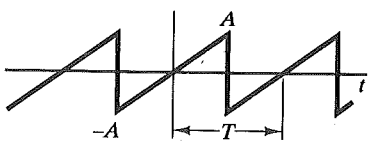
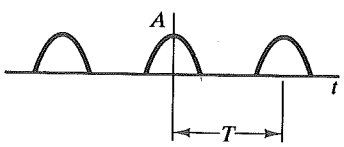
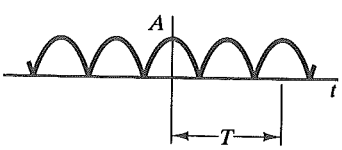
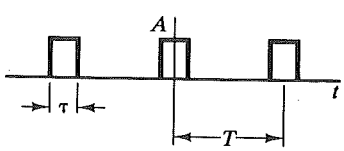
	Fourier series
Square wave 	$\frac{44}{\pi}(\cos \omega_1 t - \frac{1}{3} \cos 3\ \omega_1 t + \frac{1}{5} \cos 5\ \omega_1 t - \frac{1}{7} \cos 7\ \omega_1 t + \ldots)$
Triangular wave 	$\frac{8A}{\pi^2}(\cos \omega_1 t + \frac{1}{9} \cos 3\ \omega_1 t + \frac{1}{25} \cos 5\omega_1 t + \ldots)$
Sawtooth wave 	$\frac{24}{\pi}(\sin \omega_1 t - \frac{1}{2} \sin 2\omega_1 t + \frac{1}{3} \sin 3\omega_1 t - \frac{1}{4} \sin 4\omega_1 t + \ldots)$
Half-wave rectified cosine 	$\frac{A}{\pi}(1 + \pi \cos \omega_1 t + \frac{2}{3} \cos 2\omega_1 t - \frac{2}{15} \cos 4\omega_1 t + \frac{2}{35} \cos 6\omega_1 t \ldots (-1)\frac{n}{2} + 1 \frac{2}{n^2 - 1} \cos n\omega_1 t + \ldots)$ n even
Full-wave rectified cosine 	$\frac{2A}{\pi}(1 + 2 \cos 2\omega_1 t - \frac{2}{15} \cos 4\omega_1 t + \frac{2}{35} \cos 6\omega_1 t - \ldots (-1)\frac{n}{2} + 1 \frac{2}{n^2 - 1} \cos n\omega_1 t + \ldots)$ n even
Pulse train 	$Ad[1 + 2(\frac{\sin \pi d}{\pi d} \cos \omega_1 t + \frac{\sin 2\pi d}{2\pi d} \cos 2\omega_1 t + \frac{\sin 3\pi d}{3\pi d} \cos 3\omega_1 t + \ldots)]$ $d = \tau/T$

FIGURE 2.17 Some common periodic signals and their Fourier series.

trum consists of a continuum of frequencies. This spectrum can be defined by the Fourier transform. For a signal $x(t)$ with a spectrum $X(=f)$, the following relationships hold:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df$$

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{j2\pi ft}dt$$

Figure 2.18 presents some examples of Fourier transform pairs.

Power Spectral Density and Bandwidth

The absolute bandwidth of any time-limited signal is infinite. In practical terms, however, most of the power in a signal will be concentrated in some finite band, and the effective bandwidth will consist of that portion of the spectrum that contains most of the power. To make this concept precise, we need to define the power spectral density.

First, we observe the power in the time domain. A function $x(t)$ usually specifies a signal in terms of either voltage or current. In either case, the instantaneous power in the signal is proportional to $|x(t)|^2$. We define the average power of a time-limited signal as

$$P = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} |x(t)|^2\, dt$$

For a periodic signal, the average power in one period is

$$P = \frac{1}{T}\int_0^T |x(t)|^2\, dt$$

We would like to know the distribution of power as a function of frequency. For periodic signals, this is easily expressed in terms of the coefficients of the exponential Fourier series. The power spectral density $S(f)$ obeys

$$S(f) = \sum_{n=-\infty}^{\infty} |X_n|^2\, \delta(f - nf_0)$$

The power spectral density $S(f)$ for aperiodic functions is more difficult to define. In essence, it is obtained by defining a "period" T_0 and allowing T_0 to increase without limit.

For a continuous valued function $S(f)$, the power contained in a band of frequencies, $f_1 < f < f_2$, is

$$P = 2\int_{f_1}^{f_2} S(f)df$$

For a periodic waveform, the power through the first j harmonics is

$$P = \frac{1}{2}\sum_{n=0}^{j} |c_n|^2$$

With these concepts, we can now define the half-power bandwidth, which is perhaps the most common bandwidth definition. The half-power bandwidth is the interval between frequencies at which $S(f)$ has dropped to half of its maximum value of power, or 3 dB below the peak value.

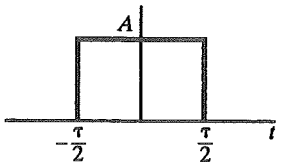
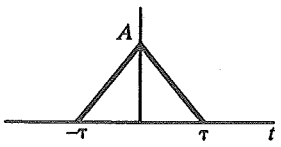
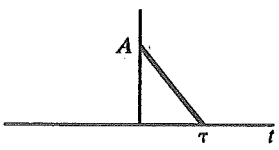
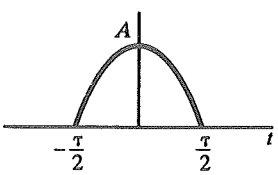
Signal $x(t)$	Spectrum X (f)
Rectangular pulse 	$A\tau \ \frac{\sin \pi f \tau}{\pi f \tau}$
Triangular pulse 	$A\tau \left(\frac{\sin \pi f \tau}{\pi f \tau}\right)^2$
Sawtooth pulse 	$\frac{jA}{2\pi f}\left[\frac{\sin \pi f \tau}{\pi f \tau} \quad e^{-2\pi f \tau} - 1\right]$
Cosine pulse 	$\frac{24\tau}{\pi} \quad \frac{\cos \pi f \tau}{1 - 4f^2\tau^2}$

FIGURE 2.18 Some common signals and their Fourier transforms.

2B APPENDIX

DECIBELS AND SIGNAL STRENGTH

AN IMPORTANT PARAMETER in any transmission system is the strength of the signal being transmitted. As a signal propagates along a transmission medium, there will be a loss, or *attenuation*, of signal strength. Additional losses occur at taps and splitters. To compensate, amplifiers may be inserted at various points to impart a gain in signal strength.

It is customary to express gains, losses, and relative levels in decibels, because

- Signal strength often falls off logarithmically, so loss is easily expressed in terms of the decibel, which is a logarithmic unit.
- The net gain or loss in a cascaded transmission path can be calculated with simple addition and subtraction.

The decibel is a measure of the difference in two signal levels:

$$N_{dB} = 10 \log_{10} \frac{P_1}{P_2}$$

where

N_{dB} = number of decibels
$P_{1,2}$ = power values
$\log_{10}$ = logarithm to the base 10 (from now on, we will simply use log to mean $\log_{10}$)

For example, if a signal with a power level of 10 *mW* is inserted onto a transmission line and the measured power some distance away is 5 *mW*, the loss can be expressed as

$$\text{LOSS} = 10 \log(5/10) = 10(-0.3) = -3 \text{ dB}$$

Note that the decibel is a measure of relative, not absolute difference. A loss from 1000 *mW* to 500 *mW* is also a –3 dB loss. Thus, a loss of 3 dB halves the voltage level; a gain of 3 dB doubles the magnitude.

The decibel is also used to measure the difference in voltage, taking into account that power is proportional to the square of the voltage:

$$P = \frac{V^2}{R}$$

where

P = power dissipated across resistance R
V = voltage across resistance R

Thus,

$$N_{dB} = 10 \log \frac{P_1}{P_2} = 10 \log \frac{V_1^2/R}{V_2^2/R} = 20 \log \frac{V_1}{V_2}$$

Decibel values refer to relative magnitudes or changes in magnitude, not to an absolute level. It is convenient to be able to refer to an absolute level of power or voltage in decibels so that gains and losses with reference to an initial signal level may easily be calculated. Thus, several derived units are in common use.

The dBW (decibel-Watt) is used extensively in microwave applications. The value of 1 *W* is selected as a reference and defined to be 0 dB*W*. The absolute decibel level of power in dB*W* is defined as

$$\text{Power(dBW)} = 10 \log \frac{\text{Power(W)}}{1\ \text{W}}$$

For example, a power of 1000 W is 30 dBW, and a power of 1 mW is –30 dBW.

A unit in common use in cable television and broadband LAN applications is the dBmV (decibel-millivolt). This is an absolute unit with 0 dBmV equivalent to 1 mV. Thus,

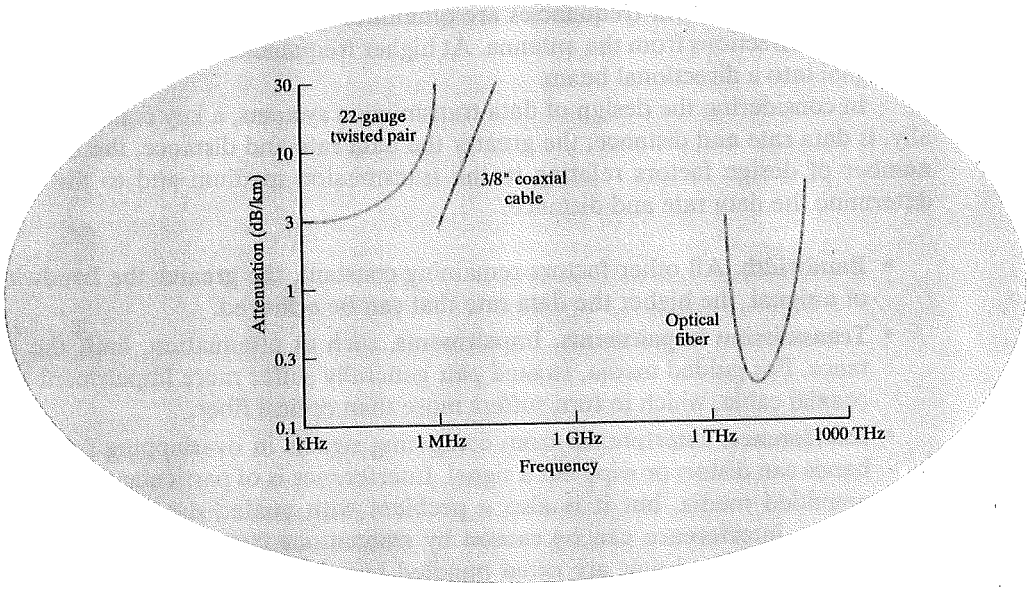
$$\text{Voltage(dBmV)} = 20 \log \frac{\text{Voltage(mV)}}{1\ \text{mV}}$$

The voltage levels are assumed to be across a 75-ohm resistance.

The decibel is convenient for determining overall gain or loss in a signal path. The amplifier gain, and the losses due to the cables, tap, and splitter are expressed in decibels. By using simple addition and subtraction, the signal level at the outlet is easily calculated. For example, consider a point-to-point link that consists of a transmission line with a single amplifier partway along. If the loss on the first portion of line is 13 dB, the gain of the amplifier is 30 dB, and the loss on the second portion of line is 40 dB, then the overall gain (loss) is $-13 + 30 - 40 = -23$ dB. If the original signal strength is –30 dBW, the received signal strength is –53 dBW.

CHAPTER 3

TRANSMISSION MEDIA



The transmission medium is the physical path between transmitter and receiver in a data transmission system. Transmission media can be classified as guided or unguided. In both cases, communication is in the form of electromagnetic waves. With guided media, the waves are guided along a solid medium, such as copper twisted pair, copper coaxial cable, and optical fiber. The atmosphere and outer space are examples of unguided media that provide a means of transmitting electromagnetic signals but do not guide them; this form of transmission is usually referred to as *wireless transmission*.

The characteristics and quality of a data transmission are determined both by the characteristics of the medium and the characteristics of the signal. In the case of guided media, the medium itself is more important in determining the limitations of transmission.

For unguided media, the bandwidth of the signal produced by the transmitting antenna is more important than the medium in determining transmission characteristics. One key property of signals transmitted by antenna is directionality. In general, signals at lower frequencies are omnidirectional; that is, the signal propagates in all directions from the antenna. At higher frequencies, it is possible to focus the signal into a directional beam.

In considering the design of data transmission systems, a key concern, generally, is data rate and distance: the greater the data rate and distance, the better. A number of design factors relating to the transmission medium and to the signal determine the data rate and distance:

- **Bandwidth.** All other factors remaining constant, the greater the bandwidth of a signal, the higher the data rate that can be achieved.
- **Transmission impairments.** Impairments, such as attenuation, limit the distance. For guided media, twisted pair generally suffer more impairment than coaxial cable, which in turn suffers more than optical fiber.
- **Interference.** Interference from competing signals in overlapping frequency bands can distort or wipe out a signal. Interference is of particular concern for unguided media, but it is also a problem with guided media. For guided media, interference can be caused by emanations from nearby cables. For example, twisted pair are often bundled together, and conduits often carry multiple cables. Interference can also be experienced from unguided transmissions. Proper shielding of a guided medium can minimize this problem.
- **Number of receivers.** A guided medium can be used to construct a point-to-point link or a shared link with multiple attachments. In the latter case, each attachment introduces some attenuation and distortion on the line, limiting distance and/or data rate.

Figure 3.1 depicts the electromagnetic spectrum and indicates the frequencies at which various guided media and unguided transmission techniques operate. In this chapter, we examine these guided and unguided alternatives. In all cases, we describe the systems physically, briefly discuss applications, and summarize key transmission characteristics.

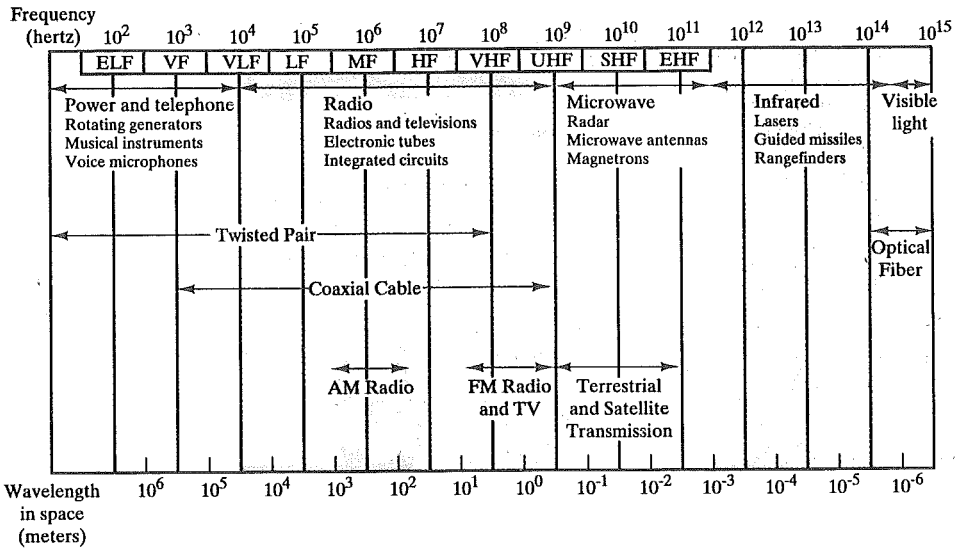


FIGURE 3.1 Electromagnetic spectrum for telecommunications.

3.1 GUIDED TRANSMISSION MEDIA

For guided transmission media, the transmission capacity, in terms of either data rate or bandwidth, depends critically on the distance and on whether the medium is point-to-point or multipoint, such as in a local area network (LAN). Table 3.1 indicates the type of performance typical for the common guided medium for long-distance point-to-point applications; we defer a discussion of the use of these media for LANs to Part II.

The three guided media commonly used for data transmission are twisted pair, coaxial cable, and optical fiber (Figure 3.2). We examine each of these in turn.

Twisted Pair

The least-expensive and most widely-used guided transmission medium is twisted pair.

TABLE 3.1 Point-to-point transmission characteristics of guided media.

Transmission medium	Total data rate	Bandwidth	Repeater spacing
Twisted pair	4 Mbps	3 MHz	2 to 10 km
Coaxial cable	500 Mbps	350 MHz	1 to 10 km
Optical fiber	2 Gbps	2 GHz	10 to 100 km

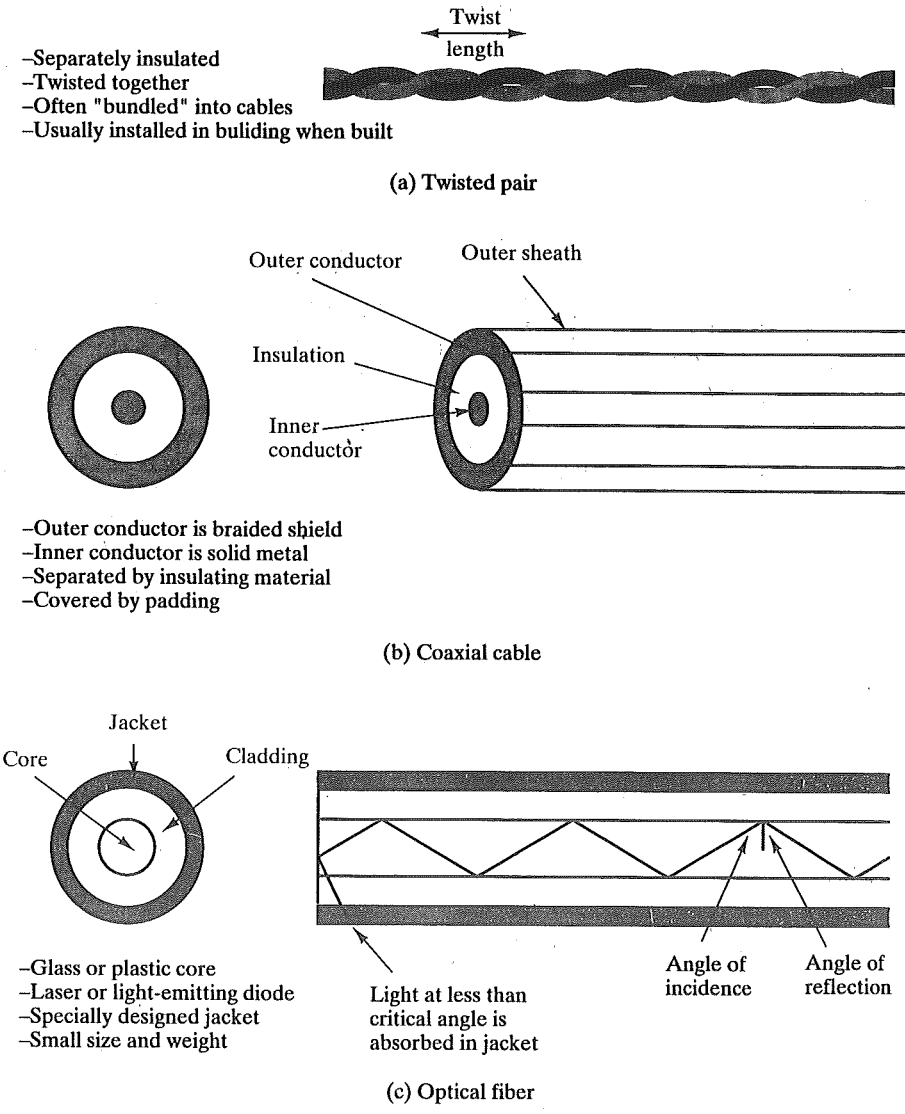


FIGURE 3.2 Guided transmission media.

Physical Description

A twisted pair consists of two insulated copper wires arranged in a regular spiral pattern. A wire pair acts as a single communication link. Typically, a number of these pairs are bundled together into a cable by wrapping them in a tough protective sheath. Over longer distances, cables may contain hundreds of pairs. The twisting tends to decrease the crosstalk interference between adjacent pairs in a cable. Neighboring pairs in a bundle typically have somewhat different twist lengths to

enhance the crosstalk interference. On long-distance links, the twist length typically varies from two to six inches. The wires in a pair have thicknesses of from 0.016 to 0.036 inches.

Applications

By far the most common transmission medium for both analog and digital signals is twisted pair. It is the most commonly used medium in the telephone network as well as being the workhorse for communications within buildings.

In the telephone system, individual residential telephone sets are connected to the local telephone exchange, or "end office," by twisted-pair wire. These are referred to as *subscriber loops*. Within an office building, each telephone is also connected to a twisted pair, which goes to the in-house private branch exchange (PBX) system or to a Centrex facility at the end office. These twisted-pair installations were designed to support voice traffic using analog signaling. However, by means of a modem, these facilities can handle digital data traffic at modest data rates.

Twisted pair is also the most common medium used for digital signaling. For connections to a digital data switch or digital PBX within a building, a data rate of 64 kbps is common. Twisted pair is also commonly used within a building for local area networks supporting personal computers. Data rates for such products are typically in the neighborhood of 10 Mbps. However, recently, twisted-pair networks with data rates of 100 Mbps have been developed, although these are quite limited in terms of the number of devices and geographic scope of the network. For long-distance applications, twisted pair can be used at data rates of 4 Mbps or more.

Twisted pair is much less expensive than the other commonly used guided transmission media (coaxial cable, optical fiber) and is easier to work with. It is more limited in terms of data rate and distance.

Transmission Characteristics

Twisted pair may be used to transmit both analog and digital signals. For analog signals, amplifiers are required about every 5 to 6 km. For digital signals, repeaters are required every 2 or 3 km.

Compared to other commonly used guided transmission media (coaxial cable, optical fiber), twisted pair is limited in distance, bandwidth, and data rate. As Figure 3.3 shows, the attenuation for twisted pair is a very strong function of frequency. Other impairments are also severe for twisted pair. The medium is quite susceptible to interference and noise because of its easy coupling with electromagnetic fields. For example, a wire run parallel to an ac power line will pick up 60-Hz energy. Impulse noise also easily intrudes into twisted pair. Several measures are taken to reduce impairments. Shielding the wire with metallic braid or sheathing reduces interference. The twisting of the wire reduces low-frequency interference, and the use of different twist lengths in adjacent pairs reduces crosstalk.

For point-to-point analog signaling, a bandwidth of up to about 250 kHz is possible. This accommodates a number of voice channels. For long-distance digital point-to-point signaling, data rates of up to a few Mbps are possible; for very short

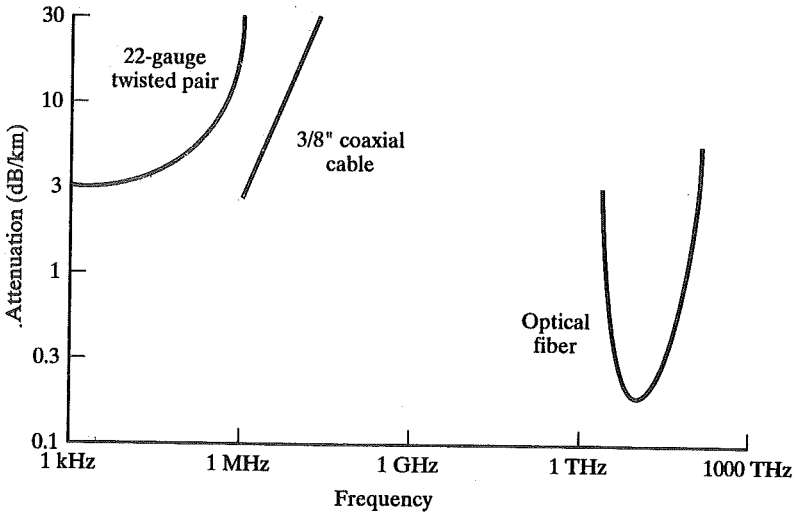


FIGURE 3.3 Attenuation of typical guided media.

distances, data rates of up to 100 Mbps have been achieved in commercially available products.

Unshielded and Shielded Twisted Pair

Twisted pair comes in two varieties: unshielded and shielded. Unshielded twisted pair (UTP) is ordinary telephone wire. Office buildings, by universal practice, are pre-wired with a great deal of excess unshielded twisted pair, more than is needed for simple telephone support. This is the least expensive of all the transmission media commonly used for local area networks, and is easy to work with and simple to install.

Unshielded twisted pair is subject to external electromagnetic interference, including interference from nearby twisted pair and from noise generated in the environment. A way to improve the characteristics of this medium is to shield the twisted pair with a metallic braid or sheathing that reduces interference. This shielded twisted pair (STP) provides better performance at lower data rates. However, it is more expensive and more difficult to work with than unshielded twisted pair.

Category 3 and Category 5 UTP

Most office buildings are prewired with a type of 100-ohm twisted pair cable commonly referred to as voice-grade. Because voice-grade twisted pair is already installed, it is an attractive alternative for use as a LAN medium. Unfortunately, the data rates and distances achievable with voice-grade twisted pair are limited.

In 1991, the Electronic Industries Association published standard EIA-568, Commercial Building Telecommunications Cabling Standard, that specified the use

of voice-grade unshielded twisted pair as well as shielded twisted pair for in-building data applications. At that time, the specification was felt to be adequate for the range of frequencies and data rates found in office environments. Up to that time, the principle interest for LAN designs was in the range of data rates from 1 Mbps to 16 Mbps. Subsequently, as users migrated to higher-performance workstations and applications, there was increasing interest in providing LANs that could operate up to 100 Mbps over inexpensive cable. In response to this need, EIA-568-A was issued in 1995. The new standard reflects advances in cable and connector design and test methods. It covers 150-ohm shielded twisted pair and 100-ohm unshielded twisted pair.

EIA-568-A recognizes three categories of UTP cabling:

- **Category 3.** UTP cables and associated connecting hardware whose transmission characteristics are specified up to 16 MHz.
- **Category 4.** UTP cables and associated connecting hardware whose transmission characteristics are specified up to 20 MHz.
- **Category 5.** UTP cables and associated connecting hardware whose transmission characteristics are specified up to 100 MHz.

Of these, it is Category 3 and Category 5 cable that have received the most attention for LAN applications. Category 3 corresponds to the voice-grade cable found in abundance in most office buildings. Over limited distances, and with proper design, data rates of up to 16 Mbps should be achievable with Category 3. Category 5 is a data-grade cable that is becoming increasingly common for preinstallation in new office buildings. Over limited distances, and with proper design, data rates of up to 100 Mbps should be achievable with Category 5.

A key difference between Category 3 and Category 5 cable is the number of twists in the cable per unit distance. Category 5 is much more tightly twisted—typically 3 to 4 twists per inch, compared to 3 to 4 twists per foot for Category 3. The tighter twisting is more expensive but provides much better performance than Category 3.

Table 3.2 summarizes the performance of Category 3 and 5 UTP, as well as the STP specified in EIA-568-A. The first parameter used for comparison, attenuation, is fairly straightforward. The strength of a signal falls off with distance over any transmission medium. For guided media, attenuation is generally logarithmic and is therefore typically expressed as a constant number of decibels per unit distance. Attenuation introduces three considerations for the designer. First, a received signal must have sufficient magnitude so that the electronic circuitry in the receiver can detect and interpret the signal. Second, the signal must maintain a level sufficiently higher than noise to be received without error. Third, attenuation is an increasing function of frequency.

Near-end crosstalk, as it applies to twisted pair wiring systems, is the coupling of the signal from one pair of conductors to another pair. These conductors may be the metal pins in a connector or the wire pairs in a cable. The near end refers to coupling that takes place when the transmit signal entering the link couples back to the

TABLE 3.2 Comparison of shielded and unshielded twisted pair.

	Attenuation (dB per 100 m)			Near-end crosstalk (dB)		
Frequency (MHz)	Category 3 UTP	Category 5 UTP	150 Ω STP	Category 3 UTP	Category 5 UTP	150 Ω STP
1	2.6	2.0	1.1	41	62	58
4	5.6	4.1	2.2	32	53	58
16	13.1	8.2	4.4	23	44	50.4
25	—	10.4	6.2	—	32	47.5
100	—	22.0	12.3	—	—	38.5
300	—	—	21.4	—	—	31.3

receive conductor pair at that same end of the link; in other words, the near-transmitted signal is picked up by the near-receive pair.

Coaxial Cable

Physical Description

Coaxial cable, like twisted pair, consists of two conductors, but is constructed differently to permit it to operate over a wider range of frequencies. It consists of a hollow outer cylindrical conductor that surrounds a single inner wire conductor (Figure 3.2b). The inner conductor is held in place by either regularly spaced insulating rings or a solid dielectric material. The outer conductor is covered with a jacket or shield. A single coaxial cable has a diameter of from 0.4 to about 1 in. Because of its shielded, concentric construction, coaxial cable is much less susceptible to interference and crosstalk than is twisted pair. Coaxial cable can be used over longer distances and supports more stations on a shared line than twisted pair.

Applications

Coaxial cable is perhaps the most versatile transmission medium and is enjoying widespread use in a wide variety of applications; the most important of these are

- Television distribution
- Long-distance telephone transmission
- Short-run computer system links
- Local area networks

Coaxial cable is spreading rapidly as a means of distributing TV signals to individual homes—cable TV. From its modest beginnings as Community Antenna Television (CATV), designed to provide service to remote areas, cable TV will eventually reach almost as many homes and offices as the telephone. A cable TV system

can carry dozens or even hundreds of TV channels at ranges up to a few tens of miles.

Coaxial cable has traditionally been an important part of the long-distance telephone network. Today, it faces increasing competition from optical fiber, terrestrial microwave, and satellite. Using frequency-division multiplexing (FDM, see Chapter 7), a coaxial cable can carry over 10,000 voice channels simultaneously.

Coaxial cable is also commonly used for short-range connections between devices. Using digital signaling, coaxial cable can be used to provide high-speed I/O channels on computer systems.

Another application area for coaxial cable is local area networks (Part Three). Coaxial cable can support a large number of devices with a variety of data and traffic types, over distances that encompass a single building or a complex of buildings.

Transmission Characteristics

Coaxial cable is used to transmit both analog and digital signals. As can be seen from Figure 3.3, coaxial cable has frequency characteristics that are superior to those of twisted pair, and can hence be used effectively at higher frequencies and data rates. Because of its shielded, concentric construction, coaxial cable is much less susceptible to interference and crosstalk than twisted pair. The principal constraints on performance are attenuation, thermal noise, and intermodulation noise. The latter is present only when several channels (FDM) or frequency bands are in use on the cable.

For long-distance transmission of analog signals, amplifiers are needed every few kilometers, with closer spacing required if higher frequencies are used. The usable spectrum for analog signaling extends to about 400 MHz. For digital signaling, repeaters are needed every kilometer or so, with closer spacing needed for higher data rates.

Optical Fiber

Physical Description

An optical fiber is a thin (2 to 125 μm), flexible medium capable of conducting an optical ray. Various glasses and plastics can be used to make optical fibers. The lowest losses have been obtained using fibers of ultrapure fused silica. Ultrapure fiber is difficult to manufacture; higher-loss multicomponent glass fibers are more economical and still provide good performance. Plastic fiber is even less costly and can be used for short-haul links, for which moderately high losses are acceptable.

An optical fiber cable has a cylindrical shape and consists of three concentric sections: the core, the cladding, and the jacket (Figure 3.2c). The *core* is the innermost section and consists of one or more very thin strands, or fibers, made of glass or plastic. Each fiber is surrounded by its own *cladding*, a glass or plastic coating that has optical properties different from those of the core. The outermost layer, surrounding one or a bundle of cladded fibers, is the *jacket*. The jacket is composed

of plastic and other material layered to protect against moisture, abrasion, crushing, and other environmental dangers.

Applications

One of the most significant technological breakthroughs in data transmission has been the development of practical fiber optic communications systems. Optical fiber already enjoys considerable use in long-distance telecommunications, and its use in military applications is growing. The continuing improvements in performance and decline in prices, together with the inherent advantages of optical fiber, have made it increasingly attractive for local area networking. The following characteristics distinguish optical fiber from twisted pair or coaxial cable:

- **Greater capacity.** The potential bandwidth, and hence data rate, of optical fiber is immense; data rates of 2 Gbps over tens of kilometers have been demonstrated. Compare this capability to the practical maximum of hundreds of Mbps over about 1 km for coaxial cable and just a few Mbps over 1 km or up to 100 Mbps over a few tens of meters for twisted pair.
- **Smaller size and lighter weight.** Optical fibers are considerably thinner than coaxial cable or bundled twisted-pair cable—at least an order of magnitude thinner for comparable information-transmission capacity. For cramped conduits in buildings and underground along public rights-of-way, the advantage of small size is considerable. The corresponding reduction in weight reduces structural support requirements.
- **Lower attenuation.** Attenuation is significantly lower for optical fiber than for coaxial cable or twisted pair (Figure 3.3) and is constant over a wide range.
- **Electromagnetic isolation.** Optical fiber systems are not affected by external electromagnetic fields. Thus, the system is not vulnerable to interference, impulse noise, or crosstalk. By the same token, fibers do not radiate energy, thereby causing little interference with other equipment and thus providing a high degree of security from eavesdropping. In addition, fiber is inherently difficult to tap.
- **Greater repeater spacing.** Fewer repeaters means lower cost and fewer sources of error. The performance of optical fiber systems from this point of view has been steadily improving. For example, AT&T has developed a fiber transmission system that achieves a data rate of 3.5 Gbps over a distance of 318 km [PARK92] without repeaters. Coaxial and twisted-pair systems generally have repeaters every few kilometers.

Five basic categories of application have become important for optical fiber:

- Long-haul trunks
- Metropolitan trunks
- Rural-exchange trunks

- Subscriber loops
- Local area networks

Long-haul fiber transmission is becoming increasingly common in the telephone network. Long-haul routes average about 900 miles in length and offer high capacity (typically 20,000 to 60,000 voice channels). These systems compete economically with microwave and have so underpriced coaxial cable in many developed countries that coaxial cable is rapidly being phased out of the telephone network in such areas.

Metropolitan trunking circuits have an average length of 7.8 miles and may have as many as 100,000 voice channels in a trunk group. Most facilities are installed in underground conduits and are repeaterless, joining telephone exchanges in a metropolitan or city area. Included in this category are routes that link long-haul microwave facilities that terminate at a city perimeter to the main telephone exchange building downtown.

Rural exchange trunks have circuit lengths ranging from 25 to 100 miles that link towns and villages. In the United States, they often connect the exchanges of different telephone companies. Most of these systems have fewer than 5,000 voice channels. The technology in these applications competes with microwave facilities.

Subscriber loop circuits are fibers that run directly from the central exchange to a subscriber. These facilities are beginning to displace twisted pair and coaxial cable links as the telephone networks evolve into full-service networks capable of handling not only voice and data, but also image and video. The initial penetration of optical fiber in this application is for the business subscriber, but fiber transmission into the home will soon begin to appear.

A final important application of optical fiber is for local area networks. Recently, standards have been developed and products introduced for optical fiber networks that have a total capacity of 100 Mbps and can support hundreds or even thousands of stations in a large office building or in a complex of buildings.

The advantages of optical fiber over twisted pair and coaxial cable become more compelling as the demand for all types of information (voice, data, image, video) increases.

Transmission Characteristics

Optical fiber systems operate in the range of about 10^{14} to 10^{15} Hz; this covers portions of the infrared and visible spectrums. The principle of optical fiber transmission is as follows. Light from a source enters the cylindrical glass or plastic core. Rays at shallow angles are reflected and propagated along the fiber; other rays are absorbed by the surrounding material. This form of propagation is called *multimode*, referring to the variety of angles that will reflect. When the fiber core radius is reduced, fewer angles will reflect. By reducing the radius of the core to the order of a wavelength, only a single angle or mode can pass: the axial ray. This single-mode propagation provides superior performance for the following reason: With multimode transmission, multiple propagation paths exist, each with a different

path length and, hence, time to traverse the fiber; this causes signal elements to spread out in time, which limits the rate at which data can be accurately received. Because there is a single transmission path with single-mode transmission, such distortion cannot occur. Finally, by varying the index of refraction of the core, a third type of transmission, known as multimode graded index, is possible. This type is intermediate between the other two in characteristics. The variable refraction has the effect of focusing the rays more efficiently than ordinary multimode, also known as multimode step index. Table 3.3 compares the three fiber transmission modes.

Two different types of light source are used in fiber optic systems: the light-emitting diode (LED) and the injection laser diode (ILD). Both are semiconductor devices that emit a beam of light when a voltage is applied. The LED is less costly, operates over a greater temperature range, and has a longer operational life. The ILD, which operates on the laser principle, is more efficient and can sustain greater data rates.

There is a relationship among the wavelength employed, the type of transmission, and the achievable data rate. Both single mode and multimode can support several different wavelengths of light and can employ laser or LED light source. In optical fiber, light propagates best in three distinct wavelength "windows," centered on 850, 1300, and 1550 nanometers (nm). These are all in the infrared portion of the frequency spectrum, below the visible-light portion, which is 400 to 700 nm. The loss is lower at higher wavelengths, allowing greater data rates over longer distances (Table 3.3). Most local applications today use 850-nm LED light sources. Although this combination is relatively inexpensive, it is generally limited to data rates under 100 Mbps and distances of a few kilometers. To achieve higher data rates and longer distances, a 1300-nm LED or laser source is needed. The highest data rates and longest distances require 1500-nm laser sources.

TABLE 3.3 Typical fiber characteristics [STER93].

Fiber type	Core diameter (μm)	Cladding diameter (μm)	Attenuation (dB/km) (Max)			Bandwidth (MHz/km) (Max)
			850 nm	1300 nm	1500 nm	
Single Mode	5.0	85 or 125	2.3			5000 @ 850 nm
	8.1	125		0.5	0.25	
Graded-index	50	125	2.4	0.6	0.5	600 @ 850 nm 1500 @ 1300 nm
	62.5	125	3.0	0.7	0.3	200 @ 850 nm 1000 @ 1300 nm
	100	140	3.5	1.5	0.9	300 @ 850 nm 500 @ 1300 nm
Step-index	200 or 300		380 or 440	6.0		6

3.2 WIRELESS TRANSMISSION

For unguided media, transmission and reception are achieved by means of an antenna. For transmission, the antenna radiates electromagnetic energy into the medium (usually air), and for reception, the antenna picks up electromagnetic waves from the surrounding medium. There are basically two types of configurations for wireless transmission: directional and omnidirectional. For the directional configuration, the transmitting antenna puts out a focused electromagnetic beam; the transmitting and receiving antennas must therefore be carefully aligned. In the omnidirectional case, the transmitted signal spreads out in all directions and can be received by many antennas. In general, the higher the frequency of a signal, the more it is possible to focus it into a directional beam.

Three general ranges of frequencies are of interest in our discussion of wireless transmission. Frequencies in the range of about 2 GHz (gigahertz = 10^9 Hz) to 40 GHz are referred to as microwave frequencies. At these frequencies, highly directional beams are possible, and microwave is quite suitable for point-to-point transmission. Microwave is also used for satellite communications. Frequencies in the range of 30 MHz to 1 GHz are suitable for omnidirectional applications. We will refer to this range as the broadcast radio range. Table 3.4 summarizes characteristics[1] of unguided transmission at various frequency bands. Microwave covers part of the UHF and all of the SHF band, and broadcast radio covers the VHF and part of the UHF band.

Another important frequency range, for local applications, is the infrared portion spectrum. This covers, roughly, from 3×10^{11} to 2×10^{14} Hz. Infrared is useful to local point-to-point and multipoint applications within confined areas, such as a single room.

Terrestrial Microwave

Physical Description

The most common type of microwave antenna is the parabolic "dish." A typical size is about 10 feet in diameter. The antenna is fixed rigidly and focuses a narrow beam to achieve line-of-sight transmission to the receiving antenna. Microwave antennas are usually located at substantial heights above ground level in order to extend the range between antennas and to be able to transmit over intervening obstacles. With no intervening obstacles, the maximum distance between antennas conforms to

$$d = 7.14\sqrt{Kh} \tag{2-1}$$

where d is the distance between antennas in kilometers, h is the antenna height in meters, and K is an adjustment factor to account for the fact that microwaves are bent or refracted with the curvature of the earth and will, hence, propagate farther

[1] The various modulation techniques are explained in Chapter 4.

TABLE 3.4 Characteristics of unguided communications bands

Frequency band	Name	Analog data		Digital data		Principal applications
		Modulation	Bandwidth	Modulation	Data rate	
30–300 kHz	LF (low frequency)	Generally not practical		ASK, FSK, MSK	0.1–100 bps	Navigation
300–3000 kHz	MF (medium frequency)	AM	To 4 kHz	ASK, FSK, MSK	10–1000 bps	Commercial AM radio
3–30 MHz	HF (high frequency)	AM, SSB	To 4 kHz	ASK, FSK, MSK	10–3000 bps	Shortwave radio CB radio
30–300 MHz	VHF (very high frequency)	AM, SSB; FM	5 kHz to 5 MHz	FSK, PSK	To 100 kbps	VHF television FM radio
300–3000 MHz	UHF (ultra high frequency)	FM, SSB	To 20 MHz	PSK	To 10 Mbps	UHF television Terrestrial microwave
3–30 GHz	SHF (super high frequency)	FM	To 500 MHz	PSK	To 100 Mbps	Terrestrial microwave Satellite microwave
30–300 GHz	EHF (extremely high frequency)	FM	To 1 GHz	PSK	To 750 Mbps	Experimental short point-to-point

than the optical line of sight. A good rule of thumb is $K = \frac{4}{3}$ [VALK93]. For example, two microwave antennas at a height of 100 m may be as far as $7.14 \times \sqrt{133} = 82$ km apart.

To achieve long-distance transmission, a series of microwave relay towers is used; point-to-point microwave links are strung together over the desired distance.

Applications

The primary use for terrestrial microwave systems is in long-haul telecommunications service, as an alternative to coaxial cable or optical fiber. The microwave facility requires far fewer amplifiers or repeaters than coaxial cable over the same distance, but requires line-of-sight transmission. Microwave is commonly used for both voice and television transmission.

Another increasingly common use of microwave is for short point-to-point links between buildings; this can be used for closed-circuit TV or as a data link between local area networks. Short-haul microwave can also be used for the so-called bypass application. A business can establish a microwave link to a long-distance telecommunications facility in the same city, bypassing the local telephone company.

Transmission Characteristics

Microwave transmission covers a substantial portion of the electromagnetic spectrum. Common frequencies used for transmission are in the range 2 to 40 GHz. The higher the frequency used, the higher the potential bandwidth and therefore the higher the potential data rate. Table 3.5 indicates bandwidth and data rate for some typical systems.

As with any transmission system, a main source of loss is attenuation. For microwave (and radio frequencies), the loss can be expressed as

$$L = 10\log\left(\frac{4\pi d}{\lambda}\right)^2 \text{ dB} \qquad (2\text{-}2)$$

where d is the distance and λ is the wavelength, in the same units. Loss varies as the square of the distance. In contrast, for twisted pair and coaxial cable, loss varies logarithmically with distance (linear in decibels). Repeaters or amplifiers, then, may be

TABLE 3.5 Typical digital microwave performance.

Band (GHz)	Bandwidth (MHz)	Data rate (Mbps)
2	7	12
6	30	90
11	40	90
18	220	274

TABLE 3.6 Principal microwave bands authorized for fixed telecommunications in the United States (1979).

Band name	Range (GHz)	Maximum channel bandwidth (MHz)	Necessary spectral efficiency (bits/Hz)	Type of service
2 GHz	1.71–1.85	—		Federal government
2 GHz	1.85–1.99	8		Private; local government
2 GHz	2.11–2.13	3.5	2	Common carrier (shared)
2 GHz	2.13–2.15	0.8/1.6		Private; local government
2 GHz	2.15–2.16	10		Private; multipoint
2 GHz	2.16–2.18	3.5	2	Common carrier
2 GHz	2.18–2.20	0.8/1.6		Private; local government
2 GHz	2.20–2.29	—		Federal government
2 GHz	2.45–2.50	0.8		Private; local government (shared)
4 GHz	3.70–4.20	20	4.5	Common carrier; satellite
6 GHz	5.925–6.425	30	3	Common carrier; satellite
6 GHz	6.525–6.875	5/10		Private; shared
7–8 GHz	7.125–8.40	—		Federal government
10 GHz	10.550–10.680	25		Private
11 GHz	10.7–11.7	50	2.25	Common carrier
12 GHz	12.2–12.7	10/20		Private; local government
13 GHz	13.2–13.25	25		Common carrier; private
14 GHz	14.4–15.25	—		Federal government
18 GHz	17.7–19.7	220		Common carrier; shared
18 GHz	18.36–19.04	50/100		Private; local government
22 GHz	21.2–23.6	50/100		Private; common carrier
31 GHz	31.0–31.2	50/100		Private; common carrier
38 GHz	36.0–38.6	—		Federal government
40 GHz	38.6–40.0	50		Private; common carrier
	Above 40.0	—		Developmental

placed farther apart for microwave systems—10 to 100 km is typical. Attenuation increases with rainfall, the effects of which become especially noticeable above 10 GHz. Another source of impairment is interference. With the growing popularity of microwave, transmission areas overlap and interference is always a danger. As a result, the assignment of frequency bands is strictly regulated.

Table 3.6 shows the authorized microwave frequency bands as regulated by the Federal Communications Commission (FCC). The most common bands for long-haul telecommunications are the 4 GHz to 6 GHz bands. With increasing congestion at these frequencies, the 11 GHz band is now coming into use. The 12 GHz band is used as a component of cable TV systems. Microwave links are used to provide TV signals to local CATV installations; the signals are then distributed to individual subscribers via coaxial cable. Higher-frequency microwave is being used for short point-to-point links between buildings; typically, the 22 GHz band is used. The higher microwave frequencies are less useful for longer distances because of increased attenuation but are quite adequate for shorter distances. In addition, at the higher frequencies, the antennas are smaller and cheaper.

Satellite Microwave

Physical Description

A communication satellite is, in effect, a microwave relay station. It is used to link two or more ground-based microwave transmitter/receivers, known as earth stations, or ground stations. The satellite receives transmissions on one frequency band (uplink), amplifies or repeats the signal, and transmits it on another frequency (downlink). A single orbiting satellite will operate on a number of frequency bands, called *transponder channels*, or simply *transponders*.

Figure 3.4 depicts, in a general way, two common configurations for satellite communication. In the first, the satellite is being used to provide a point-to-point link between two distant ground-based antennas. In the second, the satellite provides communications between one ground-based transmitter and a number of ground-based receivers.

For a communication satellite to function effectively, it is generally required that it remain stationary with respect to its position over the earth; otherwise, it would not be within the line of sight of its earth stations at all times. To remain stationary, the satellite must have a period of rotation equal to the earth's period of rotation. This match occurs at a height of 35,784 km.

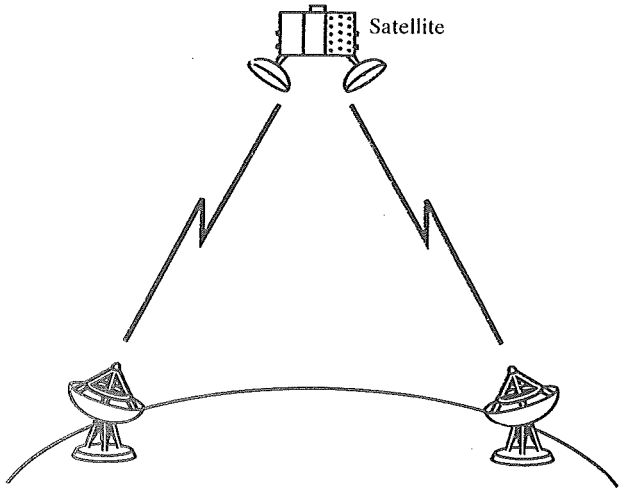
Two satellites using the same frequency band, if close enough together, will interfere with each other. To avoid this problem, current standards require a 4° spacing (angular displacement as measured from the earth) in the 4/6 GHz band and a 3° spacing at 12/14 GHz. Thus, the number of possible satellites is quite limited.

Applications

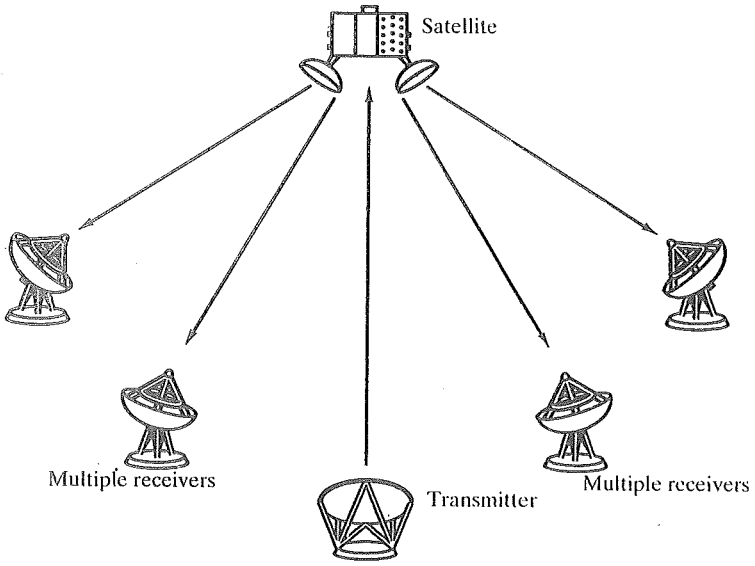
The communication satellite is a technological revolution as important as fiber optics. Among the most important applications for satellites are

- Television distribution
- Long-distance telephone transmission
- Private business networks

Because of their broadcast nature, satellites are well suited to television distribution and are being used extensively in the United States and throughout the world for this purpose. In its traditional use, a network provides programming from a central location. Programs are transmitted to the satellite and then broadcast down to a number of stations, which then distribute the programs to individual viewers. One network, the Public Broadcasting Service (PBS), distributes its television programming almost exclusively by the use of satellite channels. Other commercial networks also make substantial use of satellite, and cable television systems are receiving an ever-increasing proportion of their programming from satellites. The most recent application of satellite technology to television distribution is direct broadcast satellite (DBS), in which satellite video signals are transmitted directly to the home user. The dropping cost and size of receiving antennas have made DBS economically feasible, and a number of channels are either already in service or in the planning stage.



(a) Point-to-point link via satellite microwave



(b) Broadcast link via satellite microwave

FIGURE 3.4 Satellite communications configurations.

Satellite transmission is also used for point-to-point trunks between telephone exchange offices in public telephone networks. It is the optimum medium for high-usage international trunks and is competitive with terrestrial systems for many long-distance intranational links.

Finally, there are a number of business data applications for satellite. The satellite provider can divide the total capacity into a number of channels and lease these channels to individual business users. A user equipped with the antennas at a

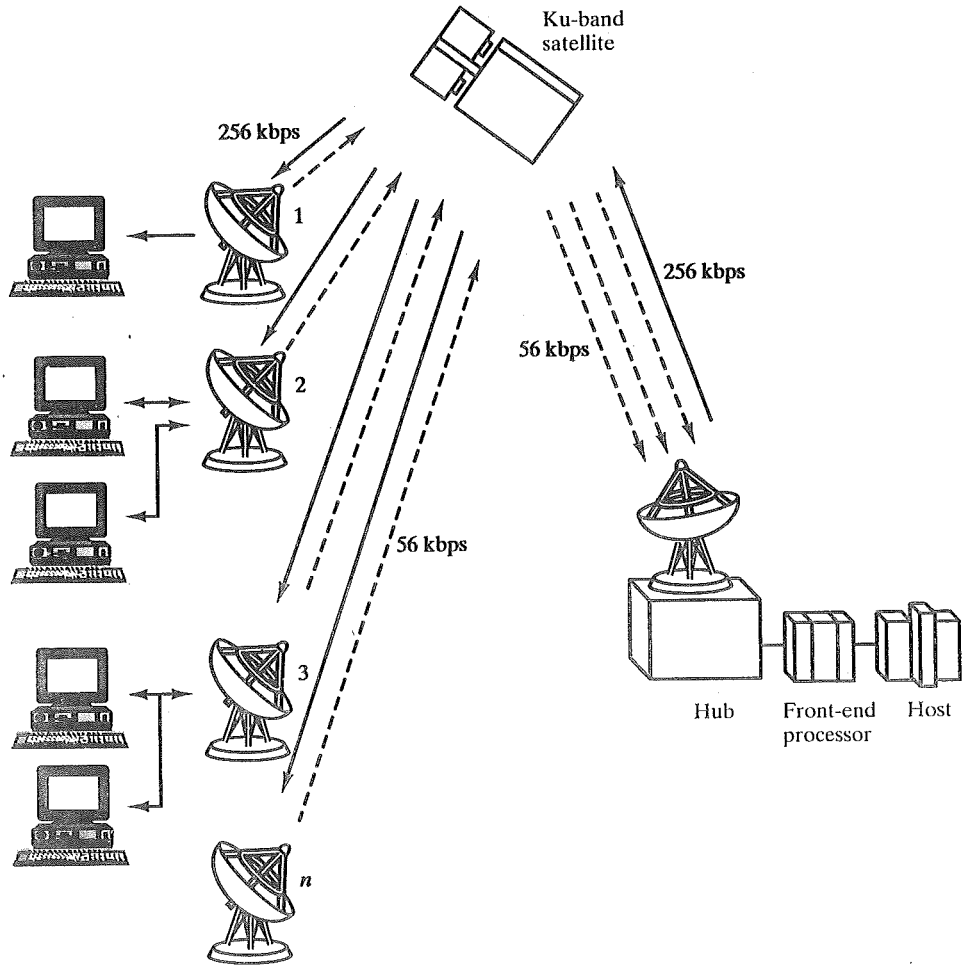


FIGURE 3.5 VSAT configuration.

number of sites can use a satellite channel for a private network. Traditionally, such applications have been quite expensive and limited to larger organizations with high-volume requirements. A recent development is the very small aperture terminal (VSAT) system, which provides a low-cost alternative. Figure 3.5 depicts a typical VSAT configuration. A number of subscriber stations are equipped with low-cost VSAT antennas (about $400 per month per VSAT). Using some protocol, these stations share a satellite transmission capacity for transmission to a hub station. The hub station can exchange messages with each of the subscribers as well as relay messages between subscribers.

Transmission Characteristics

The optimum frequency range for satellite transmission is 1 to 10 GHz. Below 1 GHz, there is significant noise from natural sources, including galactic, solar, and

atmospheric noise, and human-made interference from various electronic devices. Above 10 GHz, the signal is severely attenuated by atmospheric absorption and precipitation.

Most satellites providing point-to-point service today use a frequency bandwidth in the range 5.925 to 6.425 GHz for transmission from earth to satellite (uplink) and a bandwidth in the range 3.7 to 4.2 GHz for transmission from satellite to earth (downlink). This combination is referred to as the 4/6 GHz band. Note that the uplink and downlink frequencies differ. For continuous operation without interference, a satellite cannot transmit and receive on the same frequency. Thus, signals received from a ground station on one frequency must be transmitted back on another.

The 4/6 GHz band is within the optimum zone of 1 to 10 GHz but has become saturated. Other frequencies in that range are unavailable because of sources of interference, usually terrestrial microwave. Therefore, the 12/14 GHz band has been developed (uplink: 14 to 14.5 GHz; downlink: 11.7 to 12.2 GHz). At this frequency band, attenuation problems must be overcome. However, smaller and cheaper earth-station receivers can be used. It is anticipated that this band will also saturate, and use is projected for the 19/29 GHz band (uplink: 27.5 to 31.0 Ghz; downlink: 17.7 to 21.2 GHz). This band experiences even greater attenuation problems but will allow greater bandwidth (2500 MHz versus 500 MHz) and even smaller and cheaper receivers.

Several properties of satellite communication should be noted. First, because of the long distances involved, there is a propagation delay of about a quarter second between transmission from one earth station and reception by another earth station. This delay is noticeable in ordinary telephone conversations. It also introduces problems in the areas of error control and flow control, which we discuss in later chapters. Second, satellite microwave is inherently a broadcast facility. Many stations can transmit to the satellite, and a transmission from a satellite can be received by many stations.

Broadcast Radio

Physical Description

The principal difference between broadcast radio and microwave is that the former is omnidirectional and the latter is directional. Thus, broadcast radio does not require dish-shaped antennas, and the antennas need not be rigidly mounted to a precise alignment.

Applications

Radio is a general term used to encompass frequencies in the range of 3 kHz to 300 GHz. We are using the informal term *broadcast radio* to cover the VHF and part of the UHF band: 30 MHz to 1 GHz. This range covers FM radio as well as UHF and VHF television. This range is also used for a number of data-networking applications.

Transmission Characteristics

The range 30 MHz to 1 GHz is an effective one for broadcast communications. Unlike the case for lower-frequency electromagnetic waves, the ionosphere is trans-

parent to radio waves above 30 MHz. Transmission is limited to line of sight, and distant transmitters will not interfere with each other due to reflection from the atmosphere. Unlike the higher frequencies of the microwave region, broadcast radio waves are less sensitive to attenuation from rainfall.

As a line-of-sight propagation technique, radio obeys Equation (2-1); that is, the maximum distance between transmitter and receiver is slightly more than the optical line of sight, or 7.14 $\sqrt{Kh}$. As with microwave, the amount of attenuation due to distance obeys Equation (2-2), namely, $10\log\left(\frac{4\pi d}{\lambda}\right)^2$ dB. Because of the longer wavelength, radio waves suffer relatively less attenuation.

A prime source of impairment for broadcast radio waves is multipath interference. Reflection from land, water, and natural or human-made objects can create multiple paths between antennas. This effect is frequently evident when TV reception displays multiple images as an airplane passes by.

Infrared

Infrared communications is achieved using transmitters/receivers (transceivers) that modulate noncoherent infrared light. Transceivers must be in line of sight of each other, either directly or via reflection from a light-colored surface such as the ceiling of a room.

One important difference between infrared and microwave transmission is that the former does not penetrate walls. Thus, the security and interference problems encountered in microwave systems are not present. Furthermore, there is no frequency allocation issue with infrared, because no licensing is required.

3.3 RECOMMENDED READING

Detailed descriptions of the transmission characteristics of the transmission media discussed in this chapter can be found in [FREE91]. [REEV95] provides an excellent treatment of twisted pair and optical fiber. Two good treatments of optical fiber are [GREE93] and [STER93]. [STAL97] discusses the characteristics of transmission media for LANs in greater detail.

FREE91 Freeman, R. *Telecommunication Transmission Handbook*. New York: Wiley, 1991.

GREE93 Green, P. *Fiber Optic Networks*. Englewood Cliffs, NJ: Prentice Hall, 1993.

REEV95 Reeve, W. *Subscriber Loop Signaling and Transmission Handbook*. Piscataway, NJ: IEEE Press, 1995.

STAL97 Stallings, W. *Local and Metropolitan Area Networks, Fifth Edition*. Englewood Cliffs, NJ: Prentice Hall, 1997.

STER93 Sterling, D. *Technician's Guide to Fiber Optics*. Albany, NY: Delmar Publications, 1993.

One web site to recommend:

- http://snapple.cs.washington.edu:600/mobile/mobile_html: Source for information about wireless technology, products, conferences, and publications.

3.4 PROBLEMS

3.1 Explain the logical flaw in the following argument:

According to Table 3.1, a twisted pair can carry a digital data rate of 4 Mbps. Home computers can use a modem with the telephone network to communicate across networks. The telephone outlet is connected to the central exchange by a subscriber loop, which is twisted pair. It is difficult to establish communication by this method at a data rate higher than 28.8 kbps, which is much lower then 4 Mbps. Therefore, there must be a mistake in Table 3.1.

3.2 A twisted-pair line is approximated as a filter with the characteristics shown in Figure 3.6. The figure shows the amount of attenuation of the signal as a function of frequency. Assuming that a square wave signal, such as Figure 2.8c, with $T = 0.1$ μsec and $A = \pi/4$ is fed to the cable, find the sine wave components, with their magnitudes, that would appear at the output.

3.3 A telephone line with a bandwidth of 100 kHz is known to have a loss of 20 dB. The input signal power is measured as 0.5 watt, and the output signal noise level is measured as 2.5 μwatt. Using this information, calculate the output signal-to-noise ratio.

3.4 A transmitter-receiver pair is connected across a coaxial cable. The signal power measured at the receiver is 0.1 watt. Signal levels change 1000 times per second. Noise energy is 0.05 μJoules for every 1 millisecond. If $E_b/N_0 = 10$ dB is desired, determine how many levels must be accommodated in the signal to encode the bits. What would be the bit rate?

3.5 Given a 100-watt power source, what is the maximum allowable length for the following transmission media if a signal of 1 watt is to be received?

a. 22-gauge twisted pair operating at 1 kHz
b. 22-gauge twisted pair operating at 1 MHz
c. 0.375-inch coaxial cable operating at 1 MHz
d. 0.375-inch coaxial cable operating at 1 GHz
e. optical fiber operating at its optimal frequency

3.6 Coaxial cable is a two-wire transmission system. What is the advantage of connecting the outer conductor to ground?

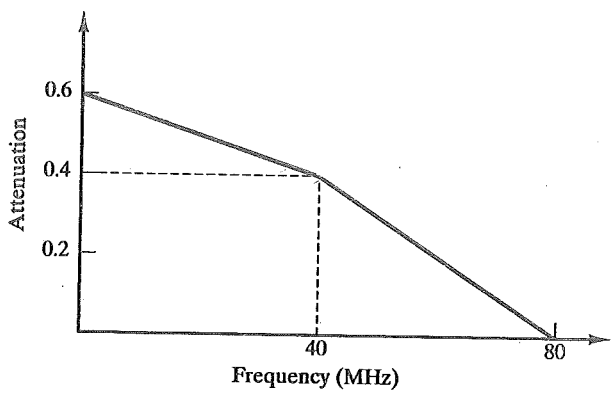
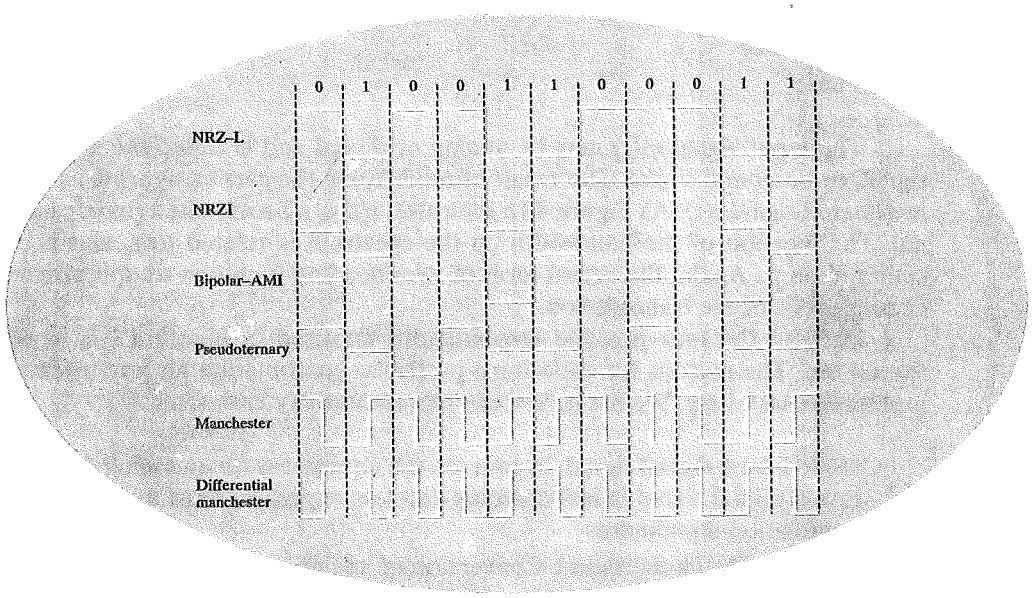


FIGURE 3.6 Filter characteristics of a twisted-pair line.

CHAPTER 4

DATA ENCODING



In Chapter 2, a distinction was made between analog and digital data, and analog and digital signals. Figure 2.13 suggested that either form of data could be encoded into either form of signal.

Figure 4.1 is another depiction that emphasizes the process involved. For digital signaling, a data source $g(t)$, which may be either digital or analog, is encoded into a digital signal $x(t)$. The actual form of $x(t)$ depends on the encoding technique, and is chosen to optimize use of the transmission medium. For example, the encoding may be chosen to either conserve bandwidth or to minimize errors.

The basis for analog signaling is a continuous, constant-frequency signal known as the carrier signal. The frequency of the carrier signal is chosen to be compatible with the transmission medium being used. Data may be transmitted using a carrier signal by modulation. Modulation is the process of encoding source data onto a carrier signal with frequency f_c. All modulation techniques involve operation on one or more of the three fundamental frequency-domain parameters:

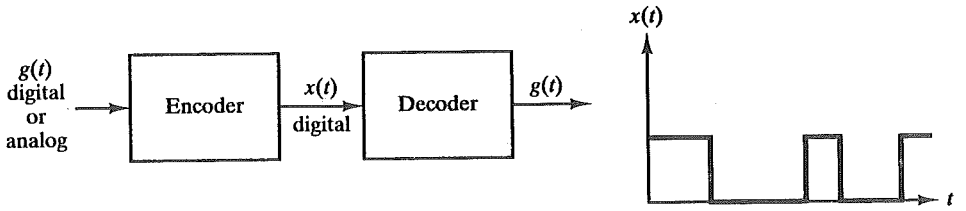
- Amplitude
- Frequency
- Phase

The input signal $m(t)$ may be analog or digital and is called the modulating signal, or baseband signal. The result of modulating the carrier signal is called the modulated signal $s(t)$. As Figure 4.1b indicates, $s(t)$ is a bandlimited (bandpass) signal. The location of the bandwidth on the spectrum is related to f_c and is often centered on f_c. Again, the actual form of the encoding is chosen to optimize some characteristic of the transmission.

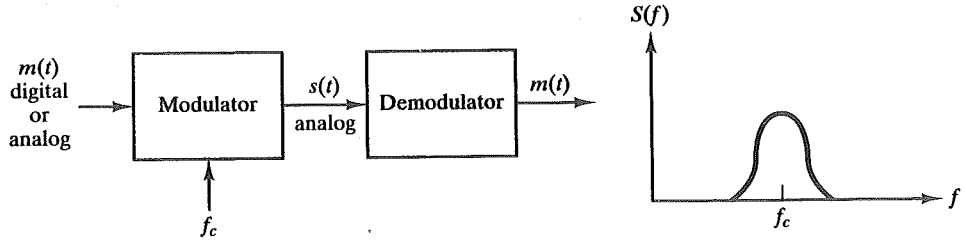
Each of the four possible combinations depicted in Figure 4.1 is in widespread use. The reasons for choosing a particular combination for any given communication task vary. We list here some representative reasons:

- **Digital data, digital signal.** In general, the equipment for encoding digital data into a digital signal is less complex and less expensive than digital-to-analog modulation equipment.
- **Analog data, digital signal.** Conversion of analog data to digital form permits the use of modern digital transmission and switching equipment. The advantages of the digital approach were outlined in Section 2.2.
- **Digital data, analog signal.** Some transmission media, such as optical fiber and the unguided media, will only propagate analog signals.
- **Analog data, analog signal.** Analog data in electrical form can be transmitted as baseband signals easily and cheaply; this is done with voice transmission over voice-grade lines. One common use of modulation is to shift the bandwidth of a baseband signal to another portion of the spectrum. In this way, multiple signals, each at a different position on the spectrum, can share the same transmission medium; this is known as frequency-division multiplexing.

We now examine the techniques involved in each of these four combinations and then look at spread spectrum, which fits into several categories.



(a) Encoding onto a digital signal



(b) Modulation onto an analog signal

FIGURE 4.1 Encoding and modulation techniques.

4.1 DIGITAL DATA, DIGITAL SIGNALS

A digital signal is a sequence of discrete, discontinuous voltage pulses. Each pulse is a signal element. Binary data are transmitted by encoding each data bit into signal elements. In the simplest case, there is a one-to-one correspondence between bits and signal elements. An example is shown in Figure 2.15, in which binary 0 is represented by a lower voltage level and binary 1 by a higher voltage level. As we shall see in this section, a variety of other encoding schemes are also used.

First, we define some terms. If the signal elements all have the same algebraic sign, that is, all positive or negative, then the signal is unipolar. In polar signaling, one logic state is represented by a positive voltage level, and the other by a negative voltage level. The data signaling rate, or just data rate, of a signal is the rate, in bits per second, that data are transmitted. The duration or length of a bit is the amount of time it takes for the transmitter to emit the bit; for a data rate R, the bit duration is 1/R. The modulation rate, in contrast, is the rate at which signal level is changed; this will depend on the nature of the digital encoding, as explained below. The modulation rate is expressed in *bauds*, which means signal elements per second. Finally, the terms *mark* and *space*, for historical reasons, refer to the binary digits 1 and 0, respectively. Table 4.1 summarizes key terms; these should be clearer when we see an example later in this section.

The tasks involved in interpreting digital signals at the receiver can be summarized by again referring to Figure 2.15. First, the receiver must know the timing

TABLE 4.1 Key data transmission terms.

Term	Units	Definition
Data element	bits	A single binary one or zero.
Data rate	bits per second (bps)	The rate at which data elements are transmitted.
Signal element	Digital: a voltage pulse of constant amplitude.	That part of a signal that occupies the shortest interval of a signaling code.
	Analog: a pulse of constant frequency, phase, and amplitude.	
Signaling rate or modulation rate	Signal elements per second (baud)	The rate at which signal elements are transmitted

of each bit. That is, the receiver must know with some accuracy when a bit begins and ends. Second, the receiver must determine whether the signal level for each bit position is high (1) or low (0). In Figure 2.15, these tasks are performed by sampling each bit position in the middle of the interval and comparing the value to a threshold. Because of noise and other impairments, there will be errors, as shown.

What factors determine how successful the receiver will be in interpreting the incoming signal? We saw in Chapter 2 that three factors are important: the signal-to-noise ratio (or, better, E_b/N_0), the data rate, and the bandwidth. With other factors held constant, the following statements are true:

- An increase in data rate increases bit error rate (the probability that a bit is received in error).
- An increase in S/N decreases bit error rate.
- An increase in bandwidth allows an increase in data rate.

There is another factor that can be used to improve performance, and that is the encoding scheme: the mapping from data bits to signal elements. A variety of encoding schemes are in use. In what follows, we describe some of the more common ones; they are defined in Table 4.2 and depicted in Figure 4.2.

Before describing these techniques, let us consider the following ways of evaluating or comparing the various techniques.

- **Signal spectrum.** Several aspects of the signal spectrum are important. A lack of high-frequency components means that less bandwidth is required for transmission. In addition, lack of a direct-current (dc) component is also desirable. With a dc component to the signal, there must be direct physical attachment of transmission components; with no dc component, ac-coupling via transformer is possible; this provides excellent electrical isolation, reducing interference. Finally, the magnitude of the effects of signal distortion and interference depend on the spectral properties of the transmitted signal. In practice, it usually happens that the transfer function of a channel is worse

TABLE 4.2 Definition of digital signal encoding formats.

Nonreturn-to-Zero-Level (NRZ-L)
0 = high level
1 = low level
Nonreturn to Zero Inverted (NRZI)
0 = no transition at beginning of interval (one bit time)
1 = transition at beginning of interval
Bipolar-AMI
0 = no line signal
1 = positive or negative level, alternating for successive ones
Pseudoternary
0 = positive or negative level, alternating for successive zeros
1 = no line signal
Manchester
0 = transition from high to low in middle of interval
1 = transition from low to high in middle of interval
Differential Manchester
Always a transition in middle of interval
0 = transition at beginning of interval
1 = no transition at beginning of interval
B8ZS
Same as bipolar AMI, except that any string of eight zeros is replaced by a string with two code violations
HDB3
Same as bipolar AMI, except that any string of four zeros is replaced by a string with one code violation

near the band edges. Therefore, a good signal design should concentrate the transmitted power in the middle of the transmission bandwidth. In such a case, a smaller distortion should be present in the received signal. To meet this objective, codes can be designed with the aim of shaping the spectrum of the transmitted signal.

- **Clocking.** We mentioned the need to determine the beginning and end of each bit position. This is no easy task. One rather expensive approach is to provide a separate clock-lead to synchronize the transmitter and receiver. The alternative is to provide some synchronization mechanism that is based on the transmitted signal; this can be achieved with suitable encoding.
- **Error detection.** We will discuss various error-detection techniques in Chapter 5, and show in Chapter 6 that these are the responsibility of a layer of logic above the signaling level known as data link control. However, it is useful to have some error-detection capability built into the physical signaling-encoding scheme; this permits errors to be detected more quickly.
- **Signal interference and noise immunity.** Certain codes exhibit superior performance in the presence of noise. This ability is usually expressed in terms of a bit error rate.
- **Cost and complexity.** Although digital logic continues to drop in price, expense should not be ignored. In particular, the higher the signaling rate to

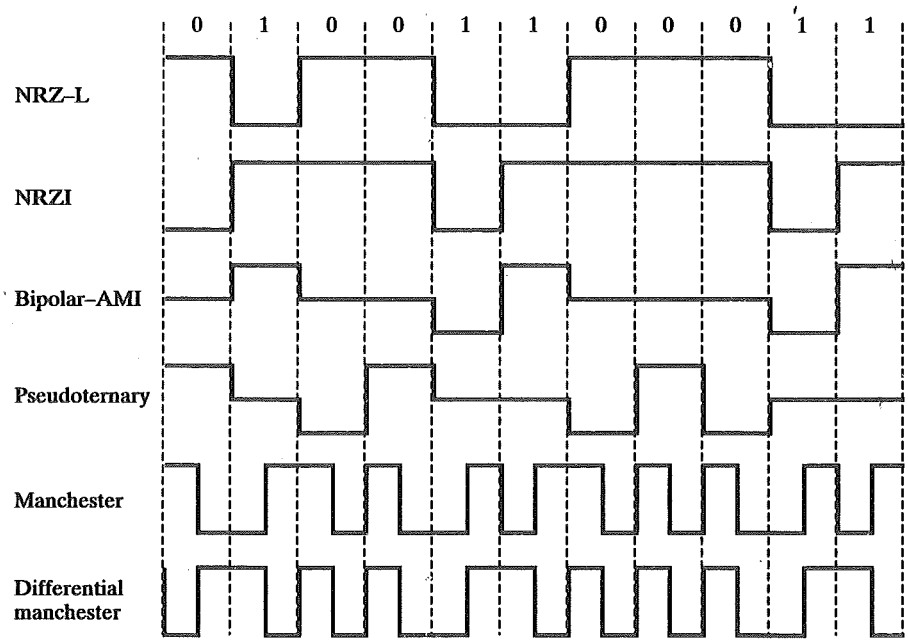


FIGURE 4.2 Digital signal encoding formats.

achieve a given data rate, the greater the cost. We will see that some codes require a signaling rate that is, in fact, greater than the actual data rate.

We now turn to a discussion of various techniques.

Nonreturn to Zero (NRZ)

The most common, and easiest, way to transmit digital signals is to use two different voltage levels for the two binary digits. Codes that follow this strategy share the property that the voltage level is constant during a bit interval; there is no transition (no return to a zero voltage level). For example, the absence of voltage can be used to represent binary 0, with a constant positive voltage used to represent binary 1. More commonly, a negative voltage is used to represent one binary value and a positive voltage is used to represent the other. This latter code, known as **Nonreturn-to-Zero-Level** (NRZ-L), is illustrated[1] in Figure 4.2. NRZ-L is generally the code used to generate or interpret digital data by terminals and other devices. If a different code is to be used for transmission, it is typically generated from an NRZ-L signal by the transmission system. (In terms of Figure 1.1, NRZ-L is $g(t)$ and the encoded signal is $s(t)$.)

A variation of NRZ is known as **NRZI** (Nonreturn to zero, invert on ones). As with NRZ-L, NRZI maintains a constant voltage pulse for the duration of a bit

[1] In this figure, a negative voltage is equated with binary 1 and a positive voltage with binary 0. This is the opposite of the definition used in virtually all other textbooks. However, there is no "standard" definition of NRZ-L, and the definition here conforms to the use of NRZ-L in data communications interfaces and the standards that govern those interfaces.

time. The data themselves are encoded as the presence or absence of a signal transition at the beginning of the bit time. A transition (low-to-high or high-to-low) at the beginning of a bit time denotes a binary 1 for that bit time; no transition indicates a binary 0.

NRZI is an example of **differential encoding**. In differential encoding, the signal is decoded by comparing the polarity of adjacent signal elements rather than determining the absolute value of a signal element. One benefit of this scheme is that it may be more reliable to detect a transition in the presence of noise than to compare a value to a threshold. Another benefit is that with a complex transmission layout, it is easy to lose the sense of the polarity of the signal. For example, on a multidrop twisted-pair line, if the leads from an attached device to the twisted pair are accidentally inverted, all 1s and 0s for NRZ-L will be inverted; this cannot happen with differential encoding.

The NRZ codes are the easiest to engineer and, in addition, make efficient use of bandwidth. This latter property is illustrated in Figure 4.3, which compares the spectral density of various encoding schemes. In the figure, frequency is normalized to the data rate. As can be seen, most of the energy in NRZ and NRZI signals is between dc and half the bit rate. For example, if an NRZ code is used to generate a signal with a data rate of 9600 bps, most of the energy in the signal is concentrated between dc and 4800 Hz.

The main limitations of NRZ signals are the presence of a dc component and the lack of synchronization capability. To picture the latter problem, consider that with a long string of 1s or 0s for NRZ-L, or a long string of 0s for NRZI, the output is a constant voltage over a long period of time. Under these circumstances, any drift between the timing of transmitter and receiver will result in a loss of synchronization between the two.

Because of their simplicity and relatively low frequency response characteristics, NRZ codes are commonly used for digital magnetic recording. However, their limitations make these codes unattractive for signal transmission applications.

Multilevel Binary

A category of encoding techniques known as multilevel-binary address some of the deficiencies of the NRZ codes. These codes use more than two signal levels. Two examples of this scheme are illustrated in Figure 4.2: bipolar-AMI (alternate mark inversion) and pseudoternary.[2]

In the case of the **bipolar-AMI** scheme, a binary 0 is represented by no line signal, and a binary 1 is represented by a positive or negative pulse. The binary 1 pulses must alternate in polarity. There are several advantages to this approach. First, there will be no loss of synchronization if a long string of 1s occurs. Each 1 introduces a transition, and the receiver can resynchronize on that transition. A long string of 0s would still be a problem. Second, because the 1 signals alternate in voltage from positive to negative, there is no net dc component. Also, the

[2] These terms are not consistently used in the literature. In some books, these two terms are used for different encoding schemes than those defined here, and a variety of terms have been used for the two schemes illustrated in Figure 4.2. The nomenclature used here corresponds to the usage in various ITU-T standards documents.

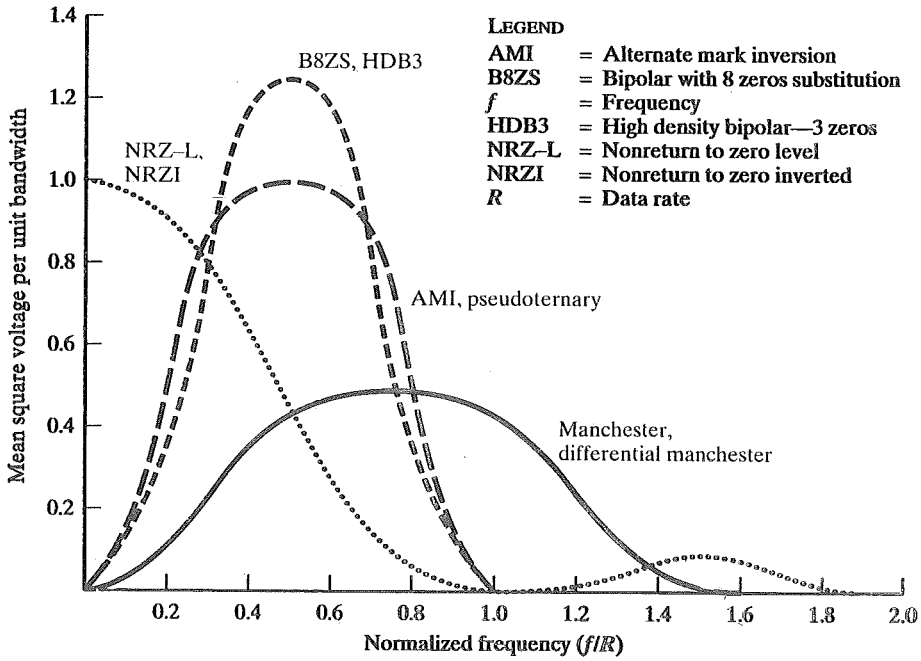


FIGURE 4.3 Spectral density of various signal encoding schemes.

bandwidth of the resulting signal is considerably less than the bandwidth for NRZ (Figure 4.3). Finally, the pulse-alternation property provides a simple means of error detection. Any isolated error, whether it deletes a pulse or adds a pulse, causes a violation of this property.

The comments of the previous paragraph also apply to pseudoternary. In this case, it is the binary 1 that is represented by the absence of a line signal, and the binary 0 by alternating positive and negative pulses. There is no particular advantage of one technique over the other, and each is the basis of some applications.

Although a degree of synchronization is provided with these codes, a long string of 0s in the case of AMI or 1s in the case of pseudoternary still presents a problem. Several techniques have been used to address this deficiency. One approach is to insert additional bits that force transitions. This technique is used in ISDN for relatively low data-rate transmission. Of course, at a high data rate, this scheme is expensive, as it results in an increase in an already high signal-transmission rate. To cope with this problem at high data rates, a technique that involves scrambling the data is used; we will look at two examples of the technique later in this section.

Thus, with suitable modification, multilevel binary schemes overcome the problems of NRZ codes. Of course, as with any engineering design decision, there is a tradeoff. With multilevel binary coding, the line signal may take on one of three levels, but each signal element, which could represent $\log_2 3 = 1.58$ bits of information, bears only one bit of information, making multilevel binary not as efficient as NRZ coding. Another way to state this is that the receiver of multilevel binary signals has to distinguish between three levels (+A, −A, 0) instead of just two levels

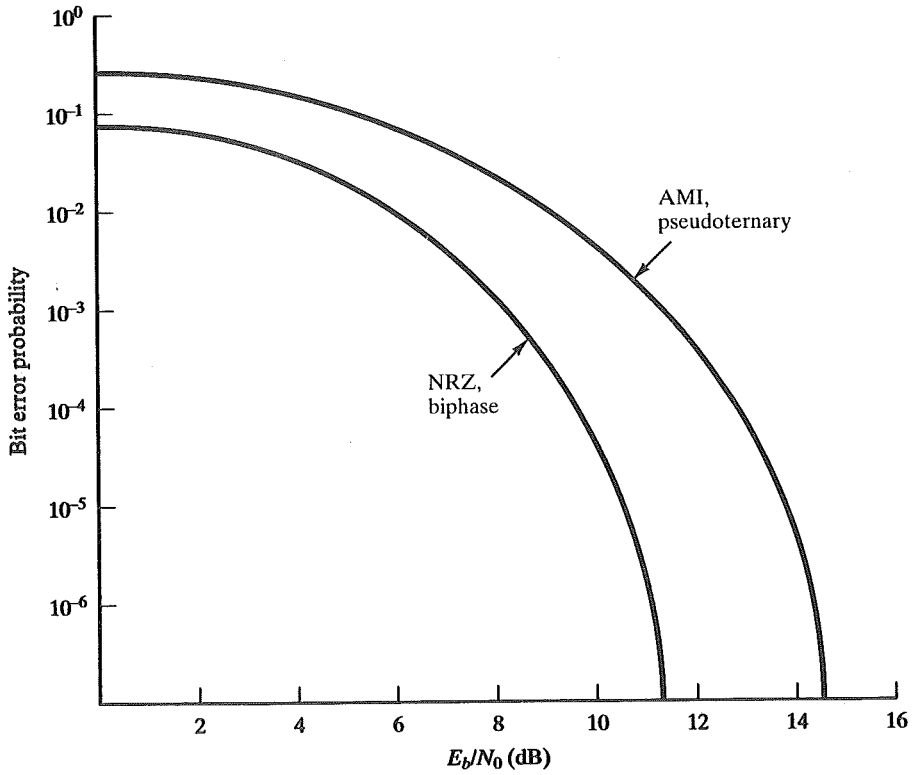


FIGURE 4.4 Theoretical bit error rate for various digital encoding schemes.

in the other signaling formats previously discussed. Because of this, the multilevel binary signal requires approximately 3 dB more signal power than a two-valued signal for the same probability of bit error; this is illustrated in Figure 4.4. Put another way, the bit error rate for NRZ codes, at a given signal-to-noise ratio, is significantly less than that for multilevel binary.

Biphase

There is another set of alternative coding techniques, grouped under the term *biphase*, which overcomes the limitations of NRZ codes. Two of these techniques, Manchester and Differential Manchester, are in common use.

In the **Manchester** code, there is a transition at the middle of each bit period. The mid-bit transition serves as a clocking mechanism and also as data: a low-to-high transition represents a 1, and a high-to-low transition represents a 0.[3] In **Dif-**

[3] The definition of Manchester presented here conforms to its usage in local area networks. In this definition, a binary 1 corresponds to a low-to-high transition, and a binary 0 to a high-to-low transition. Unfortunately, there is no official standard for Manchester, and a number of respectable textbooks (e.g., [TANE88], [COUC95], [FREE91], [SKLA88], [PEEB87], [BERT92], and the first two editions of this textbook) use the inverse, in which a low-to-high transition defines a binary 0 and a high-to-low transition defines a binary 1. Here, we conform to industry practice and to the definition used in the various LAN standards.

ferential Manchester, the mid-bit transition is used only to provide clocking. The encoding of a 0 is represented by the presence of a transition at the beginning of a bit period, and a 1 is represented by the absence of a transition at the beginning of a bit period. Differential Manchester has the added advantage of employing differential encoding.

All of the biphase techniques require at least one transition per bit time and may have as many as two transitions. Thus, the maximum modulation rate is twice that for NRZ; this means that the bandwidth required is correspondingly greater. On the other hand, the biphase schemes have several advantages:

- **Synchronization.** Because there is a predictable transition during each bit time, the receiver can synchronize on that transition. For this reason, the biphase codes are known as self-clocking codes.
- **No dc component.** Biphase codes have no dc component, yielding the benefits described earlier.
- **Error detection.** The absence of an expected transition can be used to detect errors. Noise on the line would have to invert both the signal before and after the expected transition to cause an undetected error.

As can be seen from Figure 4.3, the bulk of the energy in biphase codes is between one-half and one times the bit rate. Thus, the bandwidth is reasonably narrow and contains no dc component; however, it is wider than the bandwidth for the multilevel binary codes.

Biphase codes are popular techniques for data transmission. The more common Manchester code has been specified for the IEEE 802.3 standard for baseband coaxial cable and twisted-pair CSMA/CD bus LANs. Differential Manchester has been specified for the IEEE 802.5 token ring LAN, using shielded twisted pair.

Modulation Rate

When signal encoding techniques are used, a distinction needs to be made between data rate (expressed in bits per second), and modulation rate (expressed in baud). The data rate, or bit rate, is $1/t_B$, where t_B = bit duration. The modulation rate is the rate at which signal elements are generated. Consider, for example, Manchester encoding. The minimum size signal element is a pulse of one-half the duration of a bit interval. For a string of all binary zeroes or all binary ones, a continuous stream of such pulses is generated. Hence, the maximum modulation rate for Manchester is $2/t_B$. This situation is illustrated in Figure 4.5, which shows the transmission of a stream of 1 bits at a data rate of 1 Mbps using NRZI and Manchester. In general,

$$D = \frac{R}{b} = \frac{R}{\log_2 L}$$

where

D = modulation rate, baud
R = data rate, bps

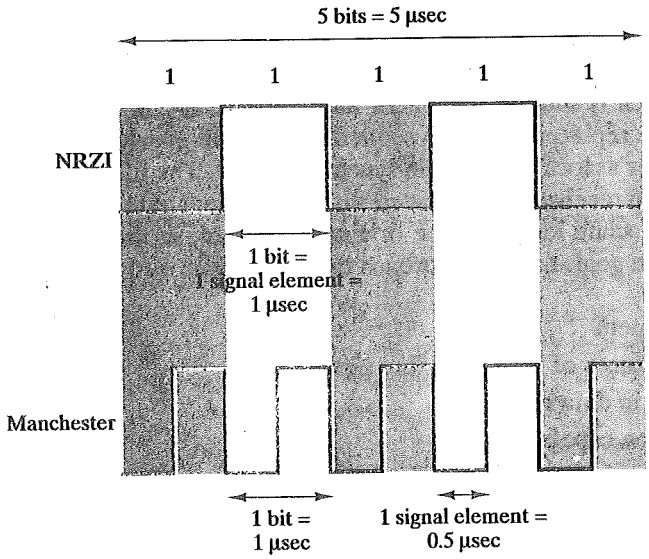


FIGURE 4.5 A stream of ones at 1 Mbps.

L = number of different signal elements
b = number of bits per signal element

One way of characterizing the modulation rate is to determine the average number of transitions that occur per bit time. In general, this will depend on the exact sequence of bits being transmitted. Table 4.3 compares transition rates for various techniques. It indicates the signal transition rate in the case of a data stream of alternating 1s and 0s, and for the data stream that produces the minimum and maximum modulation rate.

Scrambling Techniques

Although the biphase techniques have achieved widespread use in local-area-network applications at relatively high data rates (up to 10 Mbps), they have not been widely used in long-distance applications. The principal reason for this is that they

TABLE 4.3 Normalized signal transition rate of various digital signal encoding rates.

	Minimum	101010. . .	Maximum
NRZ-L	0 (all 0's or 1's)	1.0	1.0
NRZI	0 (all 0's)	0.5	1.0 (all 1's)
Binary-AMI	0 (all 0's)	1.0	1.0
Pseudoternary	0 (all 1's)	1.0	1.0
Manchester	1.0 (1010 . . .)	1.0	2.0 (all 0's or 1's)
Differential Manchester	1.0 (all 1's)	1.5	2.0 (all 0's)

require a high signaling rate relative to the data rate. This sort of inefficiency is more costly in a long-distance application.

Another approach is to make use of some sort of scrambling scheme. The idea behind this approach is simple: sequences that would result in a constant voltage level on the line are replaced by filling sequences that will provide sufficient transitions for the receiver's clock to maintain synchronization. The filling sequence must be recognized by the receiver and replaced with the original data sequence. The filling sequence is the same length as the original sequence, so there is no data-rate increase. The design goals for this approach can be summarized as follows:

- No dc component
- No long sequences of zero-level line signals
- No reduction in data rate
- Error-detection capability

Two techniques are commonly used in long-distance transmission services; these are illustrated in Figure 4.6.

A coding scheme that is commonly used in North America is known as **bipolar with 8-zeros substitution (B8ZS)**. The coding scheme is based on a bipolar-AMI. We have seen that the drawback of the AMI code is that a long string of zeros may

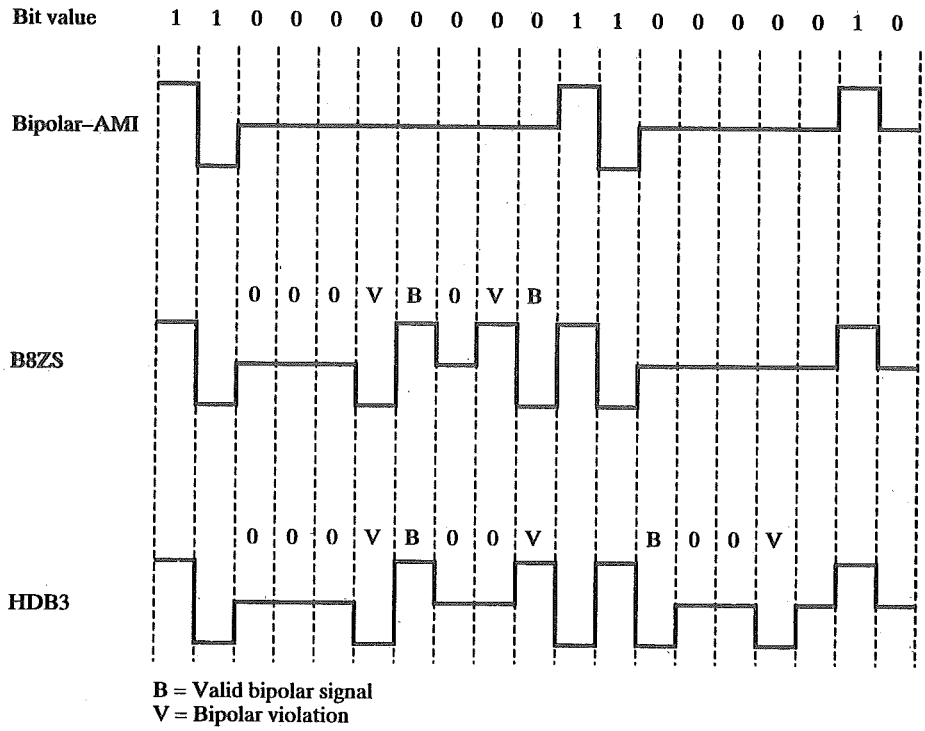


FIGURE 4.6 Encoding rules for B8ZS and HDB3.

TABLE 4.4 HDB3 substitution rules.

Polarity of preceding pulse	Number of bipolar pulses (ones) since last substitution	
	Odd	Even
−	000−	+00+
+	000+	−00−

result in loss of synchronization. To overcome this problem, the encoding is amended with the following rules:

- If an octet of all zeros occurs and the last voltage pulse preceding this octet was positive, then the eight zeros of the octet are encoded as 000+−0−+.
- If an octet of all zeros occurs and the last voltage pulse preceding this octet was negative, then the eight zeros of the octet are encoded as 000−+0+−.

This technique forces two code violations (signal patterns not allowed in AMI) of the AMI code, an event unlikely to be caused by noise or other transmission impairment. The receiver recognizes the pattern and interprets the octet as consisting of all zeros.

A coding scheme that is commonly used in Europe and Japan is known as the **high-density bipolar-3 zeros (HDB3)** code (Table 4.4). As before, it is based on the use of AMI encoding. In this case, the scheme replaces strings of four zeros with sequences containing one or two pulses. In each case, the fourth zero is replaced with a code violation. In addition, a rule is needed to ensure that successive violations are of alternate polarity so that no dc component is introduced. Thus, if the last violation was positive, this violation must be negative, and vice versa. The table shows that this condition is tested for by knowing whether the number of pulses since the last violation is even or odd and the polarity of the last pulse before the occurrence of the four zeros.

Figure 4.3 shows the spectral properties of these two codes. As can be seen, neither has a dc component. Most of the energy is concentrated in a relatively sharp spectrum around a frequency equal to one-half the data rate. Thus, these codes are well suited to high data-rate transmission.

4.2 DIGITAL DATA, ANALOG SIGNALS

We turn now to the case of transmitting digital data using analog signals. The most familiar use of this transformation is for transmitting digital data through the public telephone network. The telephone network was designed to receive, switch, and transmit analog signals in the voice-frequency range of about 300 to 3400 Hz. It is

not at present suitable for handling digital signals from the subscriber locations (although this is beginning to change). Thus, digital devices are attached to the network via a modem (modulator-demodulator), which converts digital data to analog signals, and vice versa.

For the telephone network, modems are used that produce signals in the voice-frequency range. The same basic techniques are used for modems that produce signals at higher frequencies (e.g., microwave). This section introduces these techniques and provides a brief discussion of the performance characteristics of the alternative approaches.

Encoding Techniques

We mentioned that modulation involves operation on one or more of the three characteristics of a carrier signal: amplitude, frequency, and phase. Accordingly, there are three basic encoding or modulation techniques for transforming digital data into analog signals, as illustrated in Figure 4.7:

- Amplitude-shift keying (ASK)
- Frequency-shift keying (FSK)
- Phase-shift keying (PSK)

In all these cases, the resulting signal occupies a bandwidth centered on the carrier frequency.

In ASK, the two binary values are represented by two different amplitudes of the carrier frequency. Commonly, one of the amplitudes is zero; that is, one binary digit is represented by the presence, at constant amplitude, of the carrier, the other by the absence of the carrier. The resulting signal is

$$s(t) = \begin{cases} A\cos(2\pi f_c t) & \text{binary 1} \\ 0 & \text{binary 0} \end{cases}$$

where the carrier signal is $A\cos(2\pi f_c t)$. ASK is susceptible to sudden gain changes and is a rather inefficient modulation technique. On voice-grade lines, it is typically used only up to 1200 bps.

The ASK technique is used to transmit digital data over optical fiber. For LED transmitters, the equation above is valid. That is, one signal element is represented by a light pulse while the other signal element is represented by the absence of light. Laser transmitters normally have a fixed "bias" current that causes the device to emit a low light level. This low level represents one signal element, while a higher-amplitude lightwave represents another.

In FSK, the two binary values are represented by two different frequencies near the carrier frequency. The resulting signal is

$$s(t) = \begin{cases} A\cos(2\pi f_1 t) & \text{binary 1} \\ A\cos(2\pi f_2 t) & \text{binary 0} \end{cases}$$

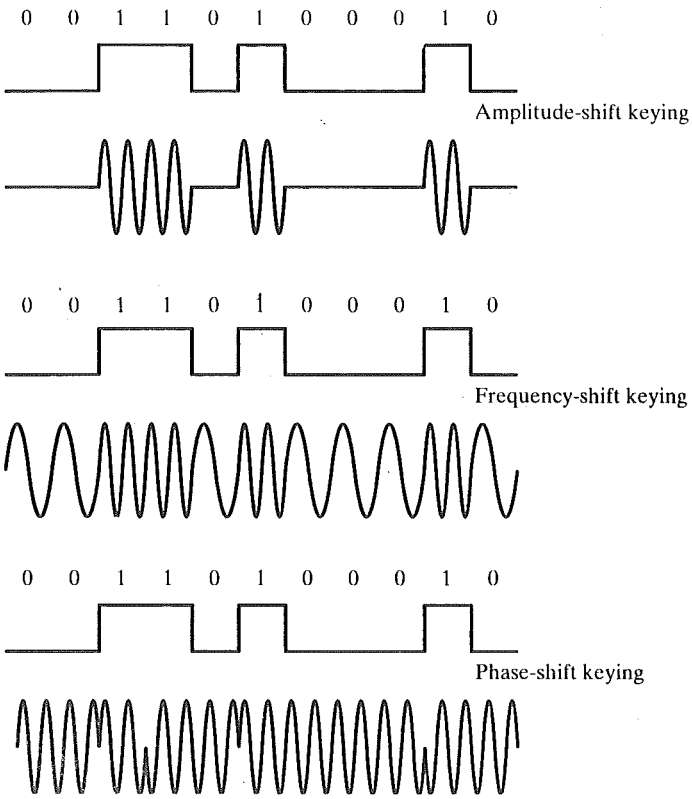


FIGURE 4.7 Modulation of analog signals for digital data.

where f_1 and f_2 are typically offset from the carrier frequency f_c by equal but opposite amounts.

Figure 4.8 shows an example of the use of FSK for full-duplex operation over a voice-grade line. The figure is a specification for the Bell System 108 series modems. Recall that a voice-grade line will pass frequencies in the approximate range of 300 to 3400 Hz, and that full-duplex means that signals are transmitted in both directions at the same time. To achieve full-duplex transmission, this bandwidth is split at 1700 Hz. In one direction (transmit or receive), the frequencies used to represent 1 and 0 are centered on 1170 Hz, with a shift of 100 Hz on either side. The effect of alternating between those two frequencies is to produce a signal whose spectrum is indicated as the shaded area on the left in Figure 4.8. Similarly, for the other direction (receive or transmit) the modem uses frequencies shifted 100 Hz to each side of a center frequency of 2125 Hz. This signal is indicated by the shaded area on the right in Figure 4.8. Note that there is little overlap and, consequently, little interference.

FSK is less susceptible to error than ASK. On voice-grade lines, it is typically used up to 1200 bps. It is also commonly used for high-frequency (3 to 30 MHz)

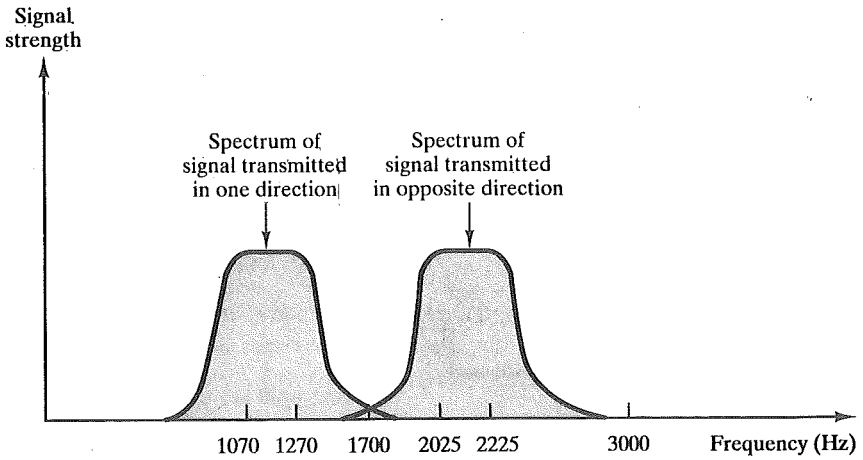


FIGURE 4.8 Full-duplex FSK transmission on a voice-grade line.

radio transmission. It can also be used at even higher frequencies on local area networks that use coaxial cable.

In PSK, the phase of the carrier signal is shifted to represent data. The bottom of Figure 4.7 is an example of a two-phase system. In this system, a binary 0 is represented by sending a signal burst of the same phase as the previous signal burst. A binary 1 is represented by sending a signal burst of opposite phase to the preceding one; this is known as differential PSK, as the phase shift is with reference to the previous bit transmitted rather than to some constant reference signal. The resulting signal is

$$s(t) = \begin{cases} A\cos(2\pi f_c t + \pi) & \text{binary 1} \\ A\cos(2\pi f_c t) & \text{binary 0} \end{cases}$$

with the phase measured relative to the previous bit interval.

More efficient use of bandwidth can be achieved if each signaling element represents more than one bit. For example, instead of a phase shift of 180<degree>, as allowed in PSK, a common encoding technique, known as quadrature-phase-shift keying (QPSK) uses phase shifts of multiples of 90°:

$$s(t) = \begin{cases} A\cos(2\pi f_c t + 45°) & 11 \\ A\cos(2\pi f_c t + 135°) & 10 \\ A\cos(2\pi f_c t + 225°) & 00 \\ A\cos(2\pi f_c t + 315°) & 01 \end{cases}$$

Thus, each signal element represents two bits rather than one.

This scheme can be extended. It is possible to transmit bits three at a time using eight different phase angles. Further, each angle can have more than one

amplitude. For example, a standard 9600 bps modem uses 12 phase angles, four of which have two amplitude values (Figure 4.9).

This latter example points out very well the difference between the data rate R (in bps) and the modulation rate D (in bauds) of a signal. Let us assume that this scheme is being employed with NRZ-L digital input. The data rate is $R = 1/t_B$ where t_B is the width of each NRZ-L bit. However, the encoded signal contains $l = 4$ bits in each signal element using $L = 16$ different combinations of amplitude and phase. The modulation rate can be seen to be R/4, as each change of signal element communicates four bits. Thus, the line signaling speed is 2400 bauds, but the data rate is 9600 bps. This is the reason that higher bit rates can be achieved over voice-grade lines by employing more complex modulation schemes.

To repeat

$$D = \frac{R}{b} = \frac{R}{\log_2 L}$$

where

D = modulation rate, bauds
R = data rate, bps
L = number of different signal elements
b = number of bits per signal element

The above is complicated when an encoding technique other than NRZ is used. For example, we saw that the maximum modulation rate for RZ signals is $2/t_B$. Thus, D for RZ is greater than D for NRZ. This, to some extent, counteracts the reduction in D achieved by using multilevel signal modulation techniques.

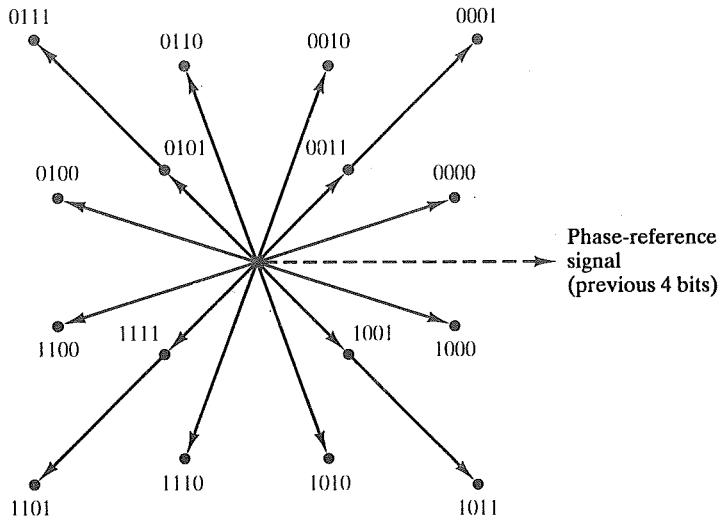


FIGURE 4.9 Phase angles for 9600 bit-per-second transmission.

Performance

In looking at the performance of various digital-to-analog modulation schemes, the first parameter of interest is the bandwidth of the modulated signal. This depends on a variety of factors, including the definition of bandwidth used and the filtering technique used to create the bandpass signal. We will use some straightforward results from [COUC95].

The transmission bandwidth B_T for ASK is of the form

$$B_T = (1 + r)R$$

where R is the bit rate and r is related to the technique by which the signal is filtered to establish a bandwidth for transmission; typically, $0 < r < 1$. The bandwidth, then, is directly related to the bit rate. The formula above is also valid for PSK.

For FSK, the bandwidth can be expressed as

$$B_T = 2\Delta F + (1 + r)R$$

where $\Delta F = f_2 - f_c = f_c - f_1$ is the offset of the modulated frequency from the carrier frequency. When very high frequencies are used, the ΔF term dominates. For example, one of the standards for FSK signaling on a coaxial cable multipoint local network uses $\Delta F = 1.25$ MHz, $f_c = 5$ MHz, and $R = 1$ Mbps. In this case, $B_T \approx 2\Delta F = 2.5$ MHz. In the example of the preceding section for the Bell 108 modem, $\Delta F = 100$ Hz, $f_c = 1170$ Hz (in one direction), and $R = 300$ bps. In this case, $B_T \approx (1 + r)R$, which is the range from 300 to 600 Hz.

With multilevel signaling, significant improvements in bandwidth can be achieved. In general

$$B_T = \left(\frac{1 + r}{l}\right)R = \left(\frac{1 + r}{\log_2 L}\right)R$$

where l is the number of bits encoded per signal element and L is the number of different signal elements.

Table 4.5 shows the ratio of data rate, R, to transmission bandwidth for various schemes. This ratio is also referred to as the bandwidth efficiency. As the name suggests, this parameter measures the efficiency with which bandwidth can be used to transmit data. The advantage of multilevel signaling methods now becomes clear.

Of course, the discussion above refers to the spectrum of the input signal to a communications line. Nothing has yet been said of performance in the presence of noise. Figure 4.10 summarizes some results based on reasonable assumptions concerning the transmission system [COUC95]. Here, bit error rate is plotted as a function of the ratio E_b/N_0 defined in Chapter 2. Of course, as that ratio increases, the bit-error rate drops. Further, PSK and QPSK are about 3 dB superior to ASK and FSK.

This information can now be related to bandwidth efficiency. Recall that

$$\frac{E_b}{N_0} = \frac{S}{N_0 R}$$

TABLE 4.5 Data rate to transmission bandwidth ratio for various digital-to-analog encoding schemes.

	$r = 0$	$r = 0.5$	$r = 1$
ASK	1.0	0.67	0.5
FSK			
Wideband ($\Delta F >> R$)	~0	~0	~0
Narrowband ($\Delta F \approx f_c$)	1.0	0.67	0.5
PSK	1.0	0.67	0.5
Multilevel signaling			
$L = 4, l = 2$	2.00	1.33	1.00
$L = 8, l = 3$	3.00	2.00	1.50
$L = 16, l = 4$	4.00	2.67	2.00
$L = 32, l = 5$	5.00	3.33	2.50

The parameter N_0 is the noise-power density in watts/hertz. Hence, the noise in a signal with bandwidth B_T is $N = N_0 B_T$. Substituting, we have

$$\frac{E_b}{N_0} = \frac{S}{N}\frac{B}{R}$$

For a given signaling scheme, the bit error rate can be reduced by increasing E_b/N_0,

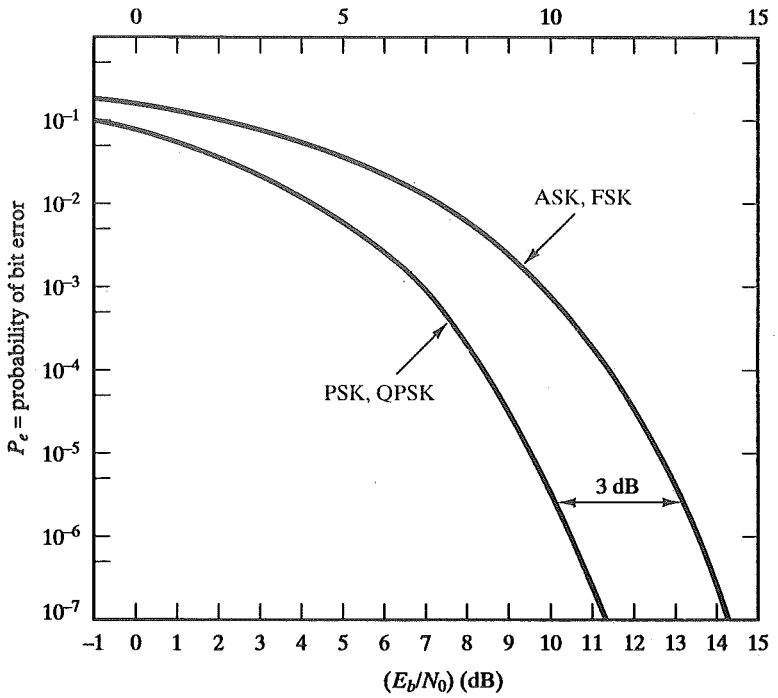


FIGURE 4.10 Bit error rate of various digital-to-analog encoding schemes.

which can be accomplished by increasing the bandwidth or decreasing the data rate—in other words, by reducing bandwidth efficiency.

Example

What is the bandwidth efficiency for FSK, ASK, PSK, and QPSK for a bit error rate of 10^{-7} on a channel with an S/N of 12 dB?

We have

$$\frac{E_b}{N_0} = 12 \text{ dB} - \left(\frac{R}{B}\right)\text{dB}$$

For FSK and ASK, from Figure 4.10,

$$\frac{E_b}{N_0} = 14.2 \text{ dB}$$

$$\left(\frac{R}{B}\right)\text{dB} = -2.2 \text{ dB}$$

$$\frac{R}{B} = 0.6$$

For PSK, from Figure 4.10

$$\frac{E_b}{N_0} = 11.2 \text{ dB}$$

$$\left(\frac{R}{B}\right)\text{dB} = -0.8 \text{ dB}$$

$$\frac{R}{B} = 1.2$$

The result for QPSK must take into account that the baud rate $D = R/2$. Thus,

$$\frac{R}{B} = 2.4$$

As the example above shows, ASK and FSK exhibit the same bandwidth efficiency; PSK is better, and even greater improvement can be achieved with multilevel signaling.

It is worthwhile to compare these bandwidth requirements with those for digital signaling. A good approximation is

$$B_T = 0.5(1 + r)D$$

where D is the modulation rate. For NRZ, $D = R$, and we have

$$\frac{R}{B} = \frac{2}{1 + r}$$

Thus, digital signaling is in the same ballpark, in terms of bandwidth efficiency, as ASK, FSK, and PSK. Significant advantage for analog signaling is seen with multilevel techniques.

4.3 ANALOG DATA, DIGITAL SIGNALS

In this section we examine the process of transforming analog data into digital signals. Strictly speaking, it might be more correct to refer to this as a process of converting analog data into digital data, a process known as digitization. Once analog data have been converted into digital data, a number of things can happen; the three most common are

1. The digital data can be transmitted using NRZ-L. In this case, we have gone directly from analog data to a digital signal.
2. The digital data can be encoded as a digital signal using a code other than NRZ-L. Thus, an extra step is required.
3. The digital data can be converted into an analog signal, using one of the modulation techniques discussed in Section 4.2.

This last, seemingly curious procedure is illustrated in Figure 4.11, which shows voice data that are digitized and then converted to an analog ASK signal; this allows digital transmission in the sense defined in Chapter 2. The voice data, because they have been digitized, can be treated as digital data, even though transmission requirements (e.g., use of microwave) dictate that an analog signal be used.

The device used for converting analog data into digital form for transmission, and subsequently recovering the original analog data from the digital, is known, as a codec (coder-decoder). In this section, we examine the two principal techniques used in codecs, pulse code modulation, and delta modulation. The section closes with a discussion of comparative performance.

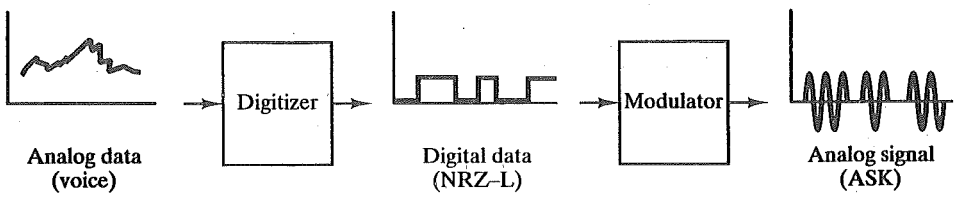


FIGURE 4.11 Digitizing analog data.

Pulse Code Modulation

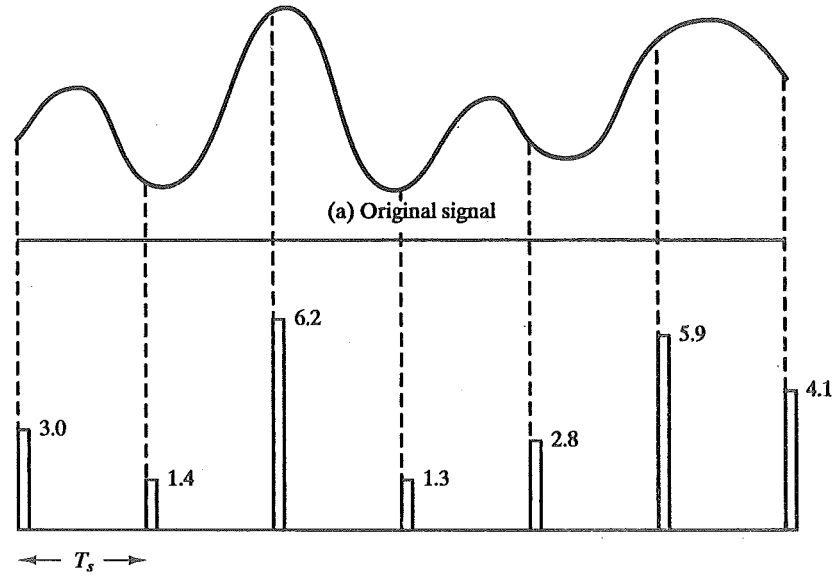
Pulse Code Modulation (PCM) is based on the sampling theorem, which states

> If a signal $f(t)$ is sampled at regular intervals of time and at a rate higher than twice the highest significant signal frequency, then the samples contain all the

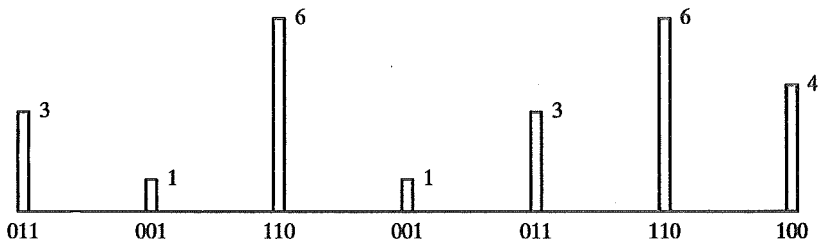
information of the original signal. The function $f(t)$ may be reconstructed from these samples by the use of a low-pass filter.

For the interested reader, a proof is provided in Appendix 4A. If voice data are limited to frequencies below 4000 Hz, a conservative procedure for intelligibility, 8000 samples per second would be sufficient to completely characterize the voice signal. Note, however, that these are analog samples.

This is illustrated in Figure 4.12a and b. The original signal is assumed to be bandlimited with a bandwidth of B. Samples are taken at a rate $2B$, or once every $1/2B$ seconds. These samples are represented as narrow pulses whose amplitude is proportional to the value of the original signal. This process is known as pulse



(b) PAM pulses



(c) PCM pulses

01100111000101111010O

(d) PCM output

FIGURE 4.12 Pulse-code modulation.

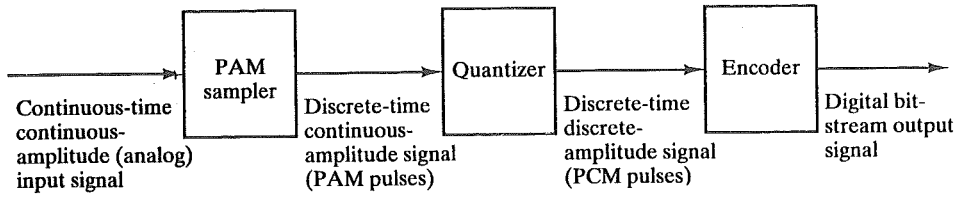


FIGURE 4.13 Analog-to-digital conversion.

amplitude modulation (PAM). By itself, this technique has commercial applicability. It is used, for example, in some of AT&T's Dimension PBX products.

However, the most significant fact about PAM is that it is the first step toward PCM, as depicted in Figure 4.12c. To produce PCM data, the PAM samples are quantized. That is, the amplitude of each PAM pulse is approximated by an n-bit integer. In the example, $n = 34$. Thus, $8 = 2^3$ levels are available for approximating the PAM pulses.

Figure 4.13 illustrates the process, starting with a continuous-time, continuous-amplitude (analog) signal, in which a digital signal is produced. The digital signal consists of blocks of n bits, where each n-bit number is the amplitude of a PCM pulse. On reception, the process is reversed to reproduce the analog signal. Notice, however, that this process violates the terms of the sampling theorem. By quantizing the PAM pulse, the original signal is now only approximated and cannot be recovered exactly. This effect is known as quantizing error or quantizing noise. The signal-to-noise ratio for quantizing noise can be expressed as

$$\frac{S}{N} = 6n - a \text{ dB}$$

where a is a constant on the order of 0 to 1. Each additional bit used for quantizing increases S/N by 6 dB, which is a factor of 4.

Typically, the PCM scheme is refined using a technique known as nonlinear encoding, which means, in effect, that the quantization levels are not equally spaced. The problem with equal spacing is that the mean absolute error for each sample is the same, regardless of signal level. Consequently, lower amplitude values are relatively more distorted. By using a greater number of quantizing steps for signals of low amplitude, and a smaller number of quantizing steps for signals of large amplitude, a marked reduction in overall signal distortion is achieved (e.g., see Figure 4.14).

The same effect can be achieved by using uniform quantizing but companding (compressing-expanding) the input analog signal. Companding is a process that compresses the intensity range of a signal by imparting more gain to weak signals than to strong signals on input. At output, the reverse operation is performed. Figure 4.15 is a typical companding function.

Nonlinear encoding can significantly improve the PCM S/N ratio. For voice signals, improvements of 24 to 30 dB have been achieved.

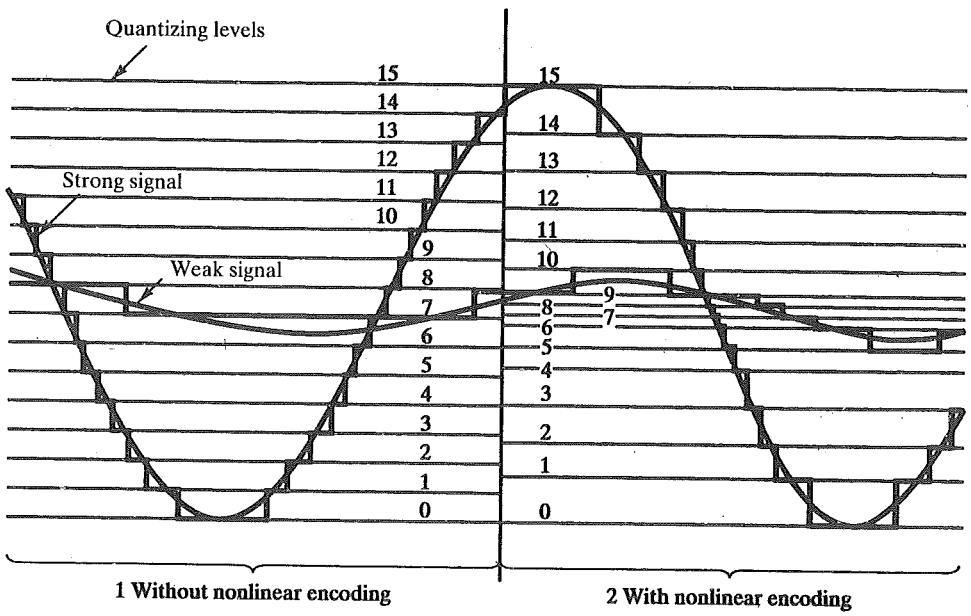


FIGURE 4.14 Effect of nonlinear coding.

Delta Modulation (DM)

A variety of techniques have been used to improve the performance of PCM or to reduce its complexity. One of the most popular alternatives to PCM is delta modulation (DM).

With delta modulation, an analog input is approximated by a staircase function that moves up or down by one quantization level (δ) at each sampling interval (T_S). An example is shown in Figure 4.16, where the staircase function is overlaid on the original analog waveform. The important characteristic of this staircase function is that its behavior is binary: At each sampling time, the function moves up or down a constant amount δ. Thus, the output of the delta modulation process can be

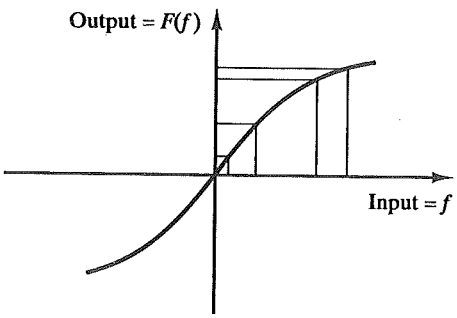


FIGURE 4.15 Typical companding function.

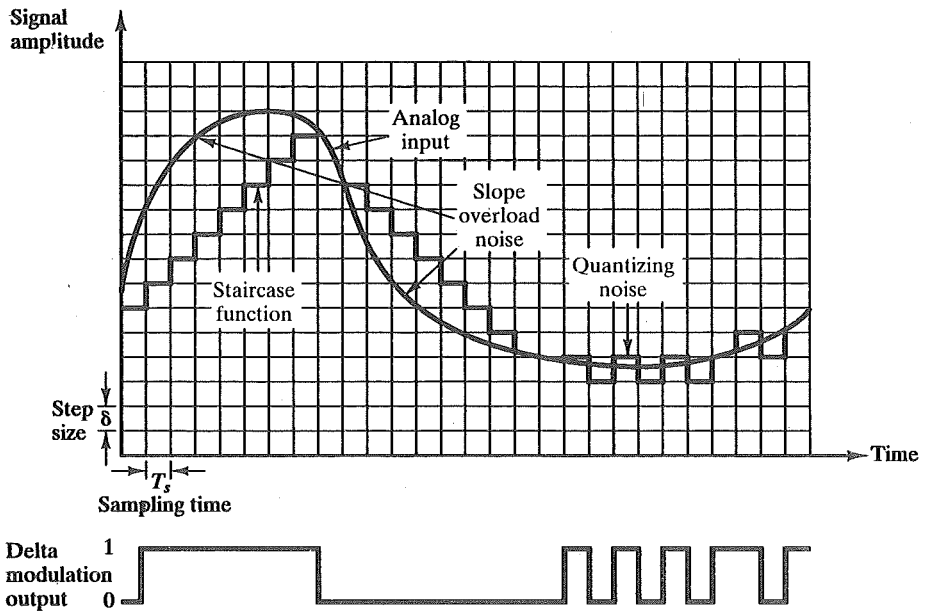


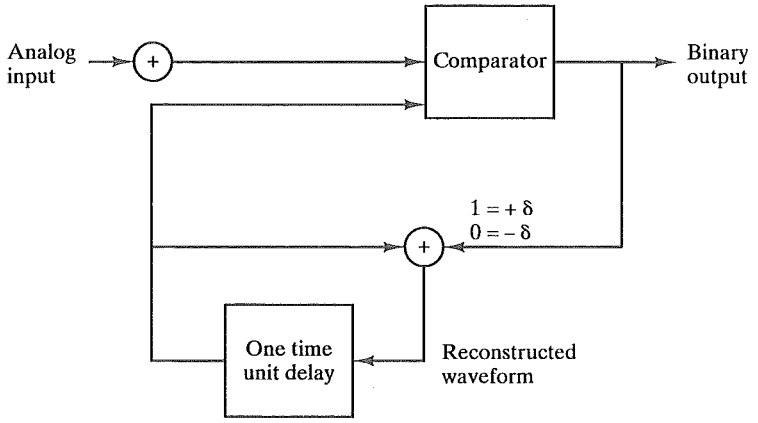
FIGURE 4.16 Example of delta modulation.

represented as a single binary digit for each sample. In essence, a bit stream is produced by approximating the derivative of an analog signal rather than its amplitude. A 1 is generated if the staircase function is to go up during the next interval; a 0 is generated otherwise.

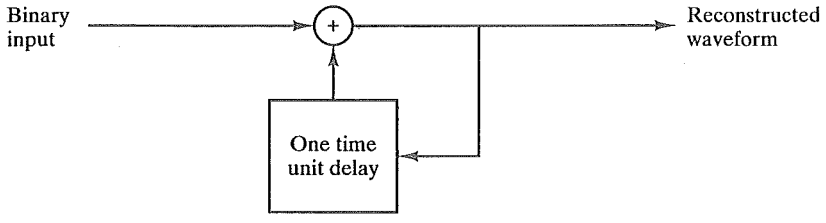
The transition (up or down) that occurs at each sampling interval is chosen so that the staircase function tracks the original analog waveform as closely as possible. Figure 4.17 illustrates the logic of the process, which is essentially a feedback mechanism. For transmission, the following occurs: At each sampling time, the analog input is compared to the most recent value of the approximating staircase function. If the value of the sampled waveform exceeds that of the staircase function, a 1 is generated; otherwise, a 0 is generated. Thus, the staircase is always changed in the direction of the input signal. The output of the DM process is therefore a binary sequence that can be used at the receiver to reconstruct the staircase function. The staircase function can then be smoothed by some type of integration process or by passing it through a low-pass filter to produce an analog approximation of the analog input signal.

There are two important parameters in a DM scheme: the size of the step assigned to each binary digit, δ, and the sampling rate. As Figure 4.16 illustrates, δ must be chosen to produce a balance between two types of errors or noise. When the analog waveform is changing very slowly, there will be quantizing noise, which increases as δ is increased. On the other hand, when the analog waveform is changing rapidly enough such that the staircase can't follow, there is slope-overload noise. This noise increases as δ is decreased.

It should be clear that the accuracy of the scheme can be improved by increasing the sampling rate; however, this increases the data rate of the output signal.



(a) Transmission



(b) Reception

FIGURE 4.17 Delta modulation.

The principal advantage of DM over PCM is the simplicity of its implementation. In general, PCM exhibits better S/N characteristics at the same data rate.

Performance

Good voice reproduction via PCM can be achieved with 128 quantization levels, or 7-bit coding ($2^7 = 128$). A voice signal, conservatively, occupies a bandwidth of 4 kHz. Thus, according to the sampling theorem, samples should be taken at a rate of 8000 per second. This implies a data rate of $8000 \times 7 = 56$ kbps for the PCM-encoded digital data.

Consider what this means from the point of view of bandwidth requirement. An analog voice signal occupies 4 kHz. A 56-kbps digital signal will require on the order of at least 28 kHz! Even more severe differences are seen with higher bandwidth signals. For example, a common PCM scheme for color television uses 10-bit codes, which works out to 92 Mbps for a 4.6-MHz bandwidth signal. In spite of these numbers, digital techniques continue to grow in popularity for transmitting analog data. The principal reasons for this are

- Because repeaters are used instead of amplifiers, there is no additive noise.

- As we shall see, time-division multiplexing (TDM) is used for digital signals instead of the frequency-division multiplexing (FDM) used for analog signals. With TDM, there is no intermodulation noise, whereas we have seen that this is a concern for FDM.
- The conversion to digital signaling allows the use of the more efficient digital switching techniques.

Furthermore, techniques are being developed to provide more efficient codes. In the case of voice, a reasonable goal appears to be in the neighborhood of 4 kbps. With video, advantage can be taken of the fact that from frame to frame, most picture elements will not change. Interframe coding techniques should allow the video requirement to be reduced to about 15 Mbps, and for slowly changing scenes, such as found in a video teleconference, down to 64 kbps or less.

As a final point, we mention that in many instances, the use of a telecommunications system will result in both digital-to-analog and analog-to-digital processing. The overwhelming majority of local terminations into the telecommunications network are analog, and the network itself uses a mixture of analog and digital techniques. As a result, digital data at a user's terminal may be converted to analog by a modem, subsequently digitized by a codec, and perhaps suffer repeated conversions before reaching its destination.

Because of the above, telecommunication facilities handle analog signals that represent both voice and digital data. The characteristics of the waveforms are quite different. Whereas voice signals tend to be skewed to the lower portion of the bandwidth (Figure 2.10), analog encoding of digital signals has a more uniform spectral content and therefore contains more high-frequency components. Studies have shown that, because of the presence of these higher frequencies, PCM-related techniques are preferable to DM-related techniques for digitizing analog signals that represent digital data.

4.4 ANALOG DATA, ANALOG SIGNALS

Modulation has been defined as the process of combining an input signal $m(t)$ and a carrier at frequency f_c to produce a signal $s(t)$ whose bandwidth is (usually) centered on f_c. For digital data, the motivation for modulation should be clear: When only analog transmission facilities are available, modulation is required to convert the digital data to analog form. The motivation when the data are already analog is less clear. After all, voice signals are transmitted over telephone lines at their original spectrum (referred to as baseband transmission). There are two principal reasons:

- A higher frequency may be needed for effective transmission. For unguided transmission, it is virtually impossible to transmit baseband signals; the required antennas would be many kilometers in diameter.
- Modulation permits frequency-division multiplexing, an important technique explored in Chapter 7.

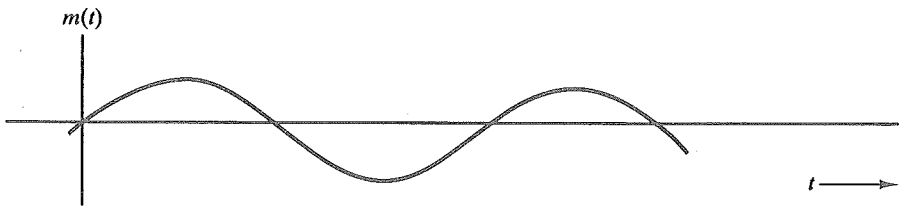
In this section, we look at the principal techniques for modulation using analog data: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). As before, the three basic characteristics of a signal are used for modulation.

Amplitude Modulation

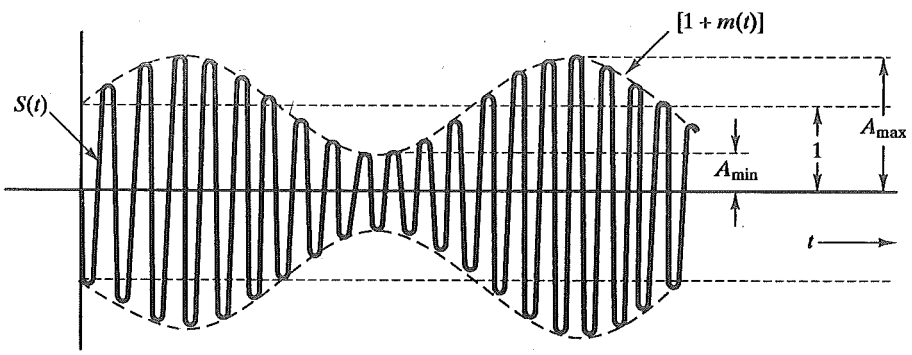
Amplitude modulation (AM) is the simplest form of modulation, and is depicted in Figure 4.18. Mathematically, the process can be expressed as

$$s(t) = [1 + n_a x(t)]\cos 2\pi f_c t$$

where $\cos 2\pi f_c t$ is the carrier and $x(t)$ is the input signal (carrying data), both normalized to unity amplitude. The parameter n_a, known as the modulation index, is the ratio of the amplitude of the input signal to the carrier. Corresponding to our previous notation, the input signal is $m(t) = n_a x(t)$. The 1 in the preceding equation is a dc component that prevents loss of information, as explained subsequently. This scheme is also known as double-sideband transmitted carrier (DSBTC).



(a) Sinusoidal modulating wave



(b) Resulting AM signal

FIGURE 4.18 Amplitude modulation.

Example

Derive an expression for $s(t)$ if $x(t)$ is the amplitude-modulating signal $\cos 2\pi f_m t$. We have

$$s(t) = [1 + n_a \cos 2\pi f_m t] \cos 2\pi f_c t$$

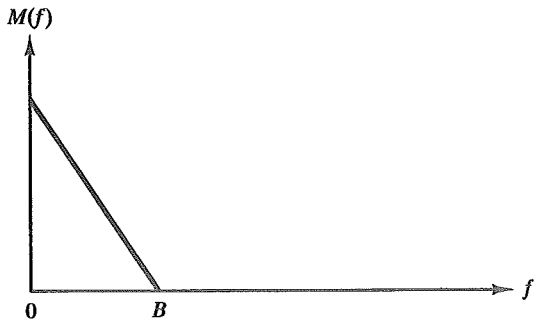
By trigonometric identity, this may be expanded to

$$s(t) = \cos 2\pi f_c t + \frac{n_a}{2} \cos 2\pi (f_c - f_m)t + \frac{n_a}{2} \cos 2\pi (f_c + f_m)t$$

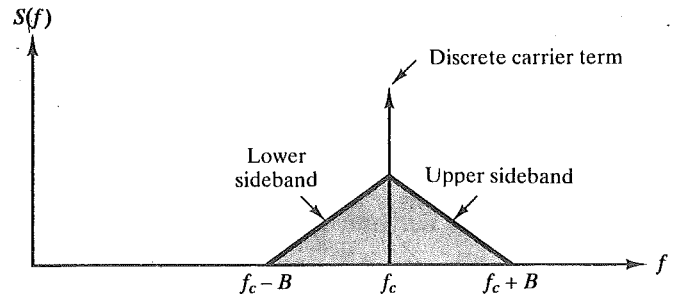
The resulting signal has a component at the original carrier frequency plus a pair of components, each spaced f_m hertz from the carrier.

From the equation above and Figure 4.18, it can be seen that AM involves the multiplication of the input signal by the carrier. The envelope of the resulting signal is $[1 + n_a x(t)]$ and, as long as $n_a < 1$, the envelope is an exact reproduction of the original signal. If $n_a > 1$, the envelope will cross the time axis and information is lost.

It is instructive to look at the spectrum of the AM signal. An example is shown in Figure 4.19. The spectrum consists of the original carrier plus the spectrum of the input signal translated to f_c. The portion of the spectrum for $|f| > |f_c|$ is the *upper sideband*, and the portion of the spectrum for $|f| < |f_c|$ is the *lower sideband*. Both the upper and lower sidebands are replicas of the original spectrum $M(f)$, with the lower sideband being frequency-reversed. As an example, consider a voice signal



(a) Spectrum of modulating signal



(b) Spectrum of AM signal with carrier at f_c

FIGURE 4.19 Spectrum of an AM signal.

with a bandwidth that extends from 300 to 3000 Hz being modulated on a 60-kHz carrier. The resulting signal contains an upper sideband of 60.3 to 63 kHz, a lower sideband of 57 to 59.7 kHz, and the 60-Hz carrier. An important relationship is

$$P_t = P_c\left(1 + \frac{n_a^2}{2}\right)$$

where P_t is the total transmitted power in $s(t)$ and P_c is the transmitted power in the carrier. We would like n_a as large as possible so that most of the signal power is used to actually carry information. However, n_a must remain below 1.

It should be clear that $s(t)$ contains unnecessary components, as each of the sidebands contains the complete spectrum of $m(t)$. A popular variant of AM, known as single sideband (SSB), takes advantage of this fact by sending only one of the sidebands, eliminating the other sideband and the carrier. The principal advantages of this approach are

- Only half the bandwidth is required; that is, $B_T = B$, where B is the bandwidth of the original signal. For DSBTC, $B_T = 2B$.
- Less power is required because no power is used to transmit the carrier or the other sideband.

Another variant is double-sideband suppressed carrier (DSBSC), which filters out the carrier frequency and sends both sidebands. This saves some power but uses as much bandwidth as DSBTC.

The disadvantage of suppressing the carrier is that the carrier can be used for synchronization purposes. For example, suppose that the original analog signal is an ASK waveform encoding digital data. The receiver needs to know the starting point of each bit time to interpret the data correctly. A constant carrier provides a clocking mechanism by which to time the arrival of bits. A compromise approach is vestigial sideband (VSB), which uses one sideband and a reduced-power carrier.

Angle Modulation

Frequency modulation (FM) and phase modulation (PM) are special cases of angle modulation. The modulated signal is expressed as

$$s(t) = A_c \cos[2\pi f_c t + \phi(t)]$$

For phase modulation, the phase is proportional to the modulating signal:

$$\phi(t) = n_p m(t)$$

where n_p is the phase modulation index.

For frequency modulation, the derivative of the phase is proportional to the modulating signal,

$$\phi'(t) = nfm(t)$$

where n_f is the frequency modulation index.

The definitions above may be clarified if we consider the following. The phase of $s(t)$ at any instant is just $2\pi f_c t + \phi(t)$. The instantaneous phase deviation from the carrier signal is $\phi(t)$. In PM, this instantaneous phase deviation is proportional to $m(t)$. Because frequency can be defined as the rate of change of phase of a signal, the instantaneous frequency of $s(t)$ is

$$2\pi f_i(t) = \frac{d}{dt}[2\pi f_c t + \phi(t)]$$

$$f_i(t) = f_c + \frac{1}{2\pi}\phi'(t)$$

and the instantaneous frequency deviation from the carrier frequency is $\phi'(t)$, which in FM is proportional to $m(t)$.

Figure 4.20 illustrates amplitude, phase, and frequency modulation by a sine wave. The shapes of the FM and PM signals are very similar. Indeed, it is impossible to tell them apart without knowledge of the modulation function.

Several observations about the FM process are in order. The peak deviation ΔF can be seen to be

$$\Delta F = \frac{1}{2\pi} n_f A_m \text{ Hz}$$

where A_m is the maximum value of $m(t)$. Thus, an increase in the magnitude of $m(t)$ will increase ΔF, which, intuitively, should increase the transmitted bandwidth B_T. However, as should be apparent from Figure 4.20, this will not increase the average power level of the FM signal, which is $A_c^2/2$; this is distinctly different from AM, where the level of modulation affects the power in the AM signal but does not affect its bandwidth.

Example

Derive an expression for $s(t)$ if $\phi(t)$ is the phase-modulating signal $n_p \cos 2\pi f_m t$. Assume that $A_c = 1$. This can be seen directly to be

$$s(t) = \cos[2\pi f_c t + n_p \cos 2\pi f_m t]$$

The instantaneous phase deviation from the carrier signal is $n_p \cos\pi f_m t$. The phase angle of the signal varies from its unmodulated value in a simple sinusoidal fashion, with the peak phase deviation equal to n_p.

The expression above can be expanded using Bessel's trigonometric identities:

$$s(t) = \sum_{n=-\infty}^{\infty} J_n(n_p)\cos\left(2\pi f_c t + 2\pi f_m t + \frac{n\pi}{2}\right)$$

where $J_n(n_p)$ is the nth order Bessel function of the first kind. Using the property

$$J_{-n}(x) = (-1)^n J_n(x)$$

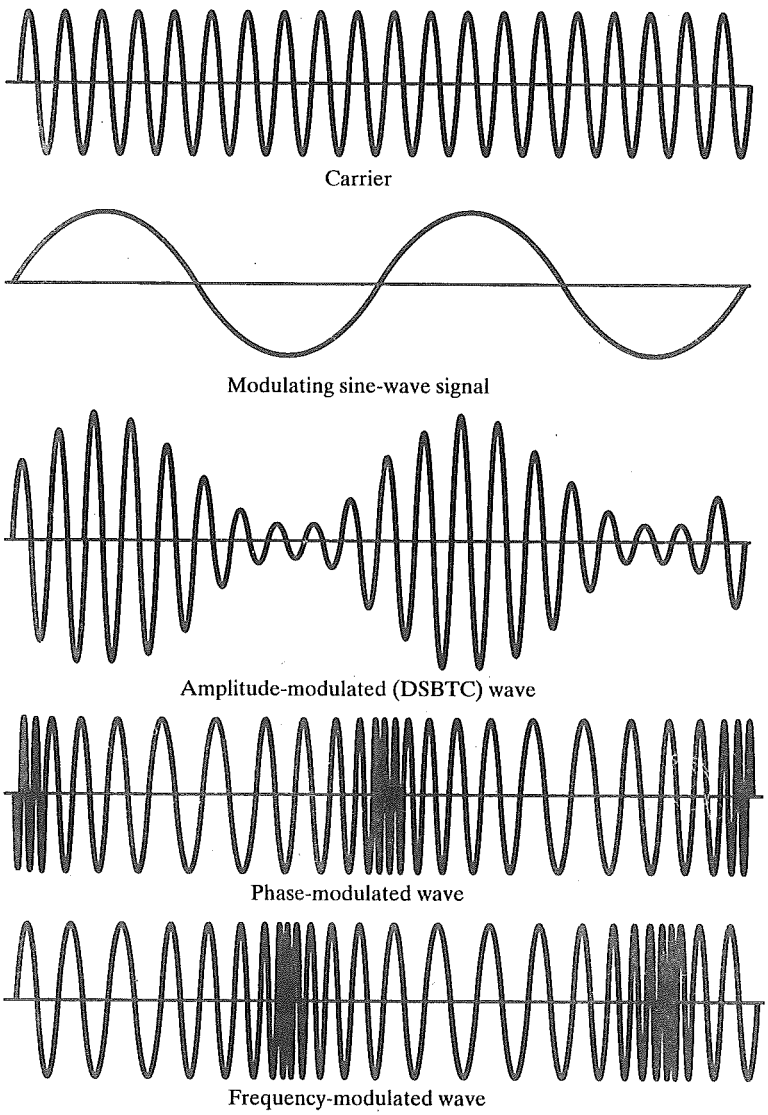


FIGURE 4.20 Amplitude, phase, and frequency modulation of a sine-wave carrier by a sine-wave signal.

this can be rewritten as

$$s(t) = J_0(n_p)\cos 2\pi f_c t + \sum_{n=-\infty}^{\infty} J_n(n_p)\left[\cos\left(2\pi(f_c + nf_m)t + \frac{n\pi}{2}\right) + \cos\left(2\pi(f_c - nf_m)t + \frac{(n+2)\pi}{2}\right)\right]$$

The resulting signal has a component at the original carrier frequency plus a set of

sidebands displaced from f_c by all possible multiples of f_m. For $n_p << 1$, the higher-order terms fall off rapidly.

Example

Derive an expression for $s(t)$ if $\phi'(t)$ is the frequency-modulating signal $-n_f \sin 2\pi f_m t$. The form of $\phi'(t)$ was chosen for convenience. We have

$$X_s(f) = \sum_{n=-\infty}^{\infty} P_n X(f - nf_s) \quad \phi(t) = -\int n_f \sin 2\pi f_m t \, dt = \frac{n_f}{2\pi f_m} \cos 2\pi f_m t$$

Thus,

$$s(t) = \cos\left[2\pi f_c t + \frac{n_f}{2\pi f_m} \cos 2\pi f_m t\right]$$

$$= \cos\left[2\pi f_c t + \frac{\Delta F}{f_m} \cos 2\pi f_m t\right]$$

The instantaneous frequency deviation from the carrier signal is $-n_f \sin 2\pi f_m t$. The frequency of the signal varies from its unmodulated value in a simple sinusoidal fashion, with the peak frequency deviation equal to n_f radians/second.

The equation for the FM signal has the identical form as for the PM signal, with $\Delta F/f_m$ substituted for n_p. Thus, the Bessel expansion is the same.

As with AM, both FM and PM result in a signal whose bandwidth is centered at f_c. However, we can now see that the magnitude of that bandwidth is very different. Amplitude modulation is a linear process and produces frequencies that are the sum and difference of the carrier signal and the components of the modulating signal. Hence, for AM

$$B_T = 2B$$

However, angle modulation includes a term of the form $\cos(\phi(t))$, which is nonlinear and will produce a wide range of frequencies. In essence, for a modulating sinusoid of frequency f_m, $s(t)$ will contain components at $f_c + f_m$, $f_c + 2f_m$, and so on. In the most general case, infinite bandwidth is required to transmit an FM or PM signal. As a practical matter, a very good rule of thumb, known as Carson's rule [COUC95], is

$$B_T = 2(\beta + 1)B$$

where

$$\beta = \begin{cases} n_p A_m & \text{for PM} \\ \dfrac{\Delta F}{B} = \dfrac{n_f A_m}{2\pi B} & \text{for FM} \end{cases}$$

We can rewrite the formula for FM as

$$B_T = 2\Delta F + 2B$$

Thus, both FM and PM require greater bandwidth than AM.

4.5 SPREAD SPECTRUM

An increasingly popular form of communications is known as spread spectrum. This technique does not fit neatly into the categories defined in this chapter, as it can be used to transmit either analog or digital data, using an analog signal.

The spread spectrum technique was developed initially for military and intelligence requirements. The essential idea is to spread the information signal over a wider bandwidth in order to make jamming and interception more difficult. The first type of spread spectrum developed became known as frequency-hopping.[4] A more recent version is direct-sequence spread spectrum. Both of these techniques are used in various wireless data-network products. They also find use in other communications applications, such as cordless telephones.

Figure 4.21 highlights the key characteristics of any spread spectrum system. Input is fed into a channel encoder that produces an analog signal with a relatively narrow bandwidth around some center frequency. This signal is further modulated using a sequence of seemingly random digits known as a pseudorandom sequence. The effect of this modulation is to significantly increase the bandwidth (spread the spectrum) of the signal to be transmitted. On the receiving end, the same digit sequence is used to demodulate the spread spectrum signal. Finally, the signal is fed into a channel decoder to recover the data.

A comment about pseudorandom numbers is in order. These numbers are generated by an algorithm using some initial value called the *seed*. The algorithm is deterministic and therefore produces sequences of numbers that are not statistically random. However, if the algorithm is good, the resulting sequences will pass many reasonable tests of randomness. Such numbers are often referred to as pseudorandom numbers.[5] The important point is that unless you know the algorithm and

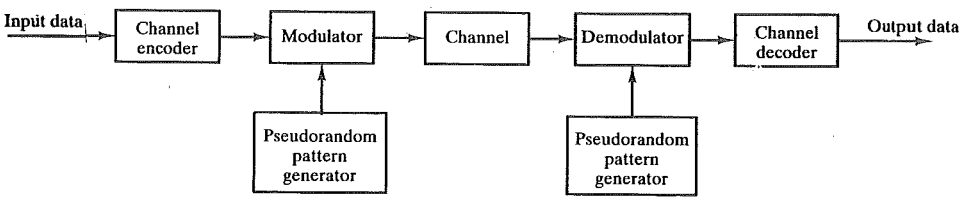


FIGURE 4.21 General model of spread spectrum digital communication system.

[4] Spread spectrum (using frequency-hopping) was invented, believe it or not, by Hollywood screen siren Hedy Lamarr in 1940 at the age of 26. She and a partner who later joined her effort were granted a patent in 1942 (U.S. Patent 2,292,387; 11 August 1942). Lamarr considered this her contribution to the war effort and never profited from her invention. For an interesting account, see [MEEK90].

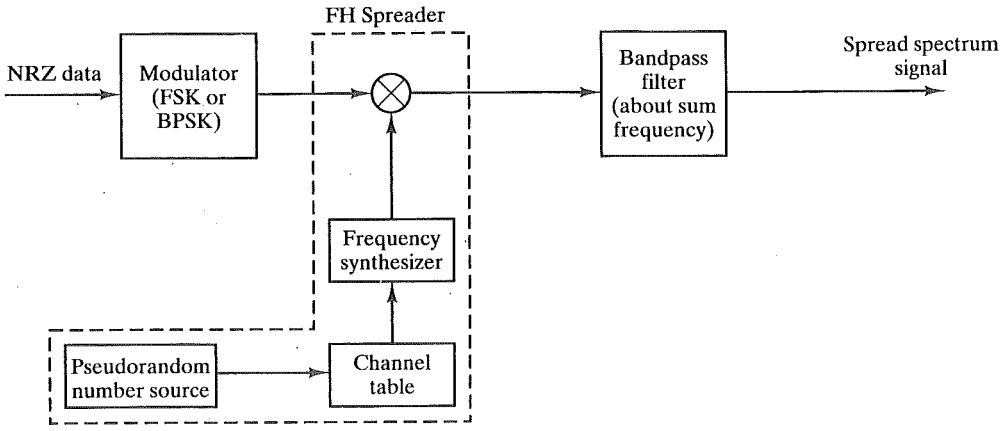
[5] See [STAL95b] for a more detailed discussion of pseudorandom numbers.

the seed, it is impractical to predict the sequence. Hence, only a receiver that shares this information with a transmitter will be able to successfully decode the signal.

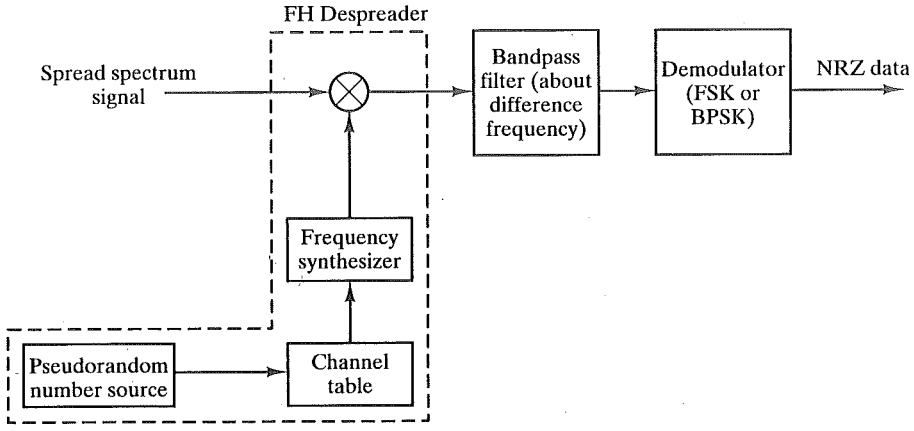
Frequency-Hopping

Under this scheme, the signal is broadcast over a seemingly random series of radio frequencies, hopping from frequency to frequency at split-second intervals. A receiver, hopping between frequencies in synchronization with the transmitter, picks up the message. Would-be eavesdroppers hear only unintelligible blips. Attempts to jam the signal succeed only at knocking out a few bits of it.

A typical block diagram for a frequency-hopping system is shown in Figure 4.22. For transmission, binary data is fed into a modulator using some digital-to-



(a) Transmitter



(b) Receiver

FIGURE 4.22 Frequency-hopping spread spectrum system.

analog encoding scheme, such as frequency-shift keying (FSK) or binary-phase shift keying (BPSK). The resulting signal is centered around some base frequency. A pseudorandom number source serves as an index into a table of frequencies. At each successive interval, a new frequency is selected from the table. This frequency is then modulated by the signal produced from the initial modulator to produce a new signal with the same shape but now centered on the frequency chosen from the table.

On reception, the spread-spectrum signal is demodulated using the same sequence of table-derived frequencies and then demodulated to produce the output data.

For example, if FSK is employed, the modulator selects one of two frequencies, say f_0 or f_c, corresponding to the transmission of binary 0 or 1. The resulting binary FSK signal is translated in frequency by an amount determined by the output sequence from the pseudorandom number generator. Thus, if the frequency selected at time i is f_i, then the signal at time i is either $f_i + f_0$ or $f_i + f_1$.

Direct Sequence

Under this scheme, each bit in the original signal is represented by multiple bits in the transmitted signal, known as a chipping code. The chipping code spreads the signal across a wider frequency band in direct proportion to the number of bits used. Therefore, a 10-bit chipping code spreads the signal across a frequency band that is 10 times greater than a 1-bit chipping code.

One technique with direct-sequence spread spectrum is to combine the digital information stream with the pseudorandom bit stream using an exclusive-or. Figure

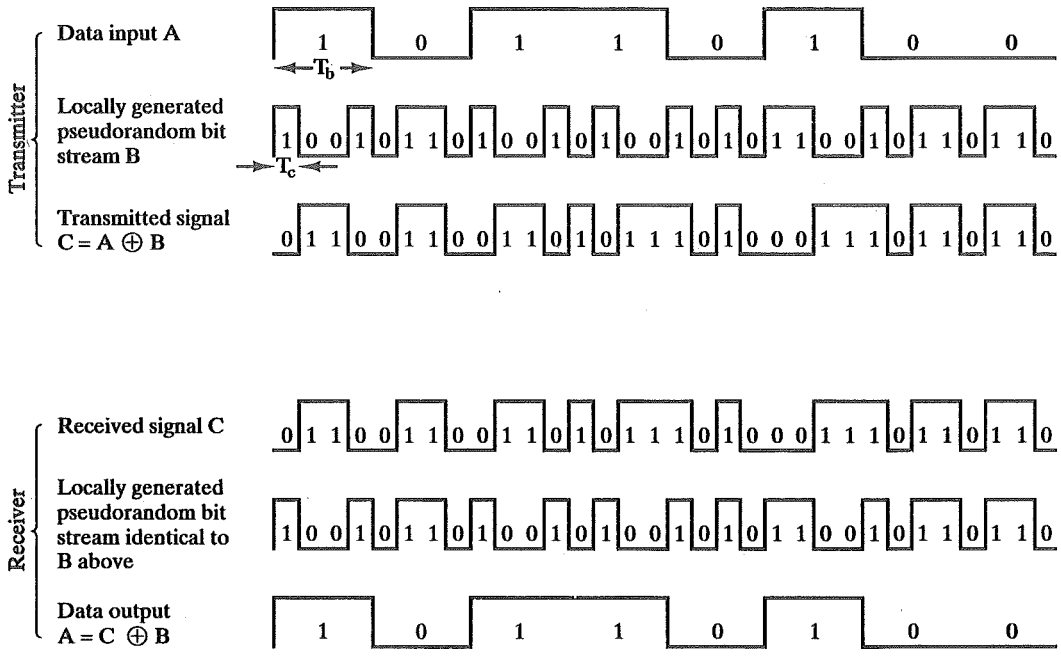
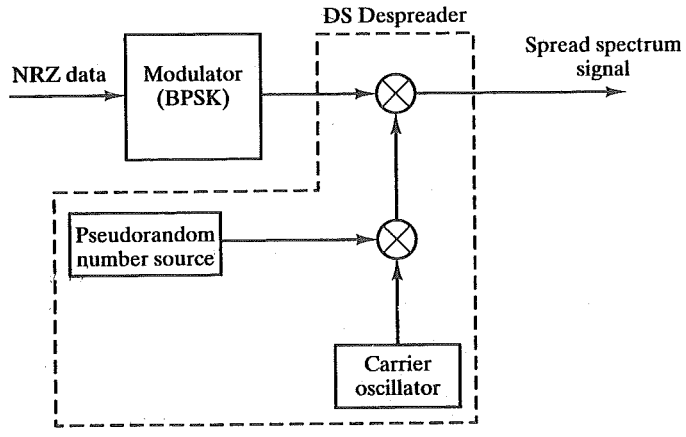
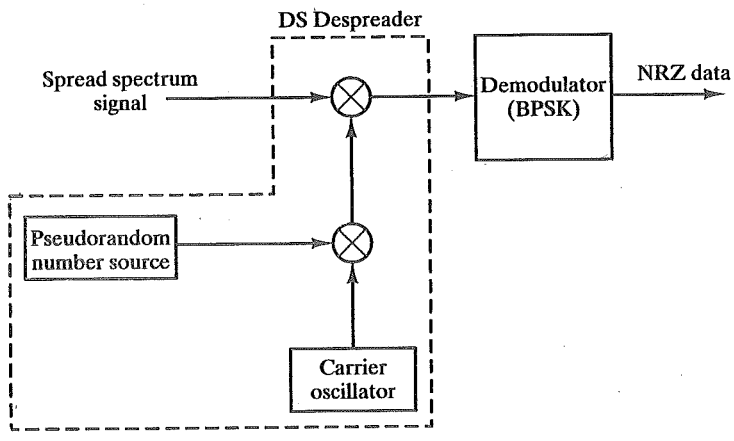


FIGURE 4.23 Example of direct sequence spread spectrum.



(a) Transmitter



(b) Receiver

FIGURE 4.24 Direct sequence spread spectrum system.

4.23 shows an example. Note that an information bit of 1 inverts the pseudorandom bits in the combination, while an information bit of 0 causes the pseudorandom bits to be transmitted without inversion. The combination bit stream has the data rate of the original pseudorandom sequence, so it has a wider bandwidth than the information stream. In this example, the pseudorandom bit stream is clocked at four times the information rate.

Figure 4.24 shows a typical direct sequence implementation. In this case, the information stream and the pseudorandom stream are both converted to analog signals and then combined, rather than performing the exclusive-or of the two streams and then modulating.

The spectrum spreading achieved by the direct sequence technique is easily determined. For example, suppose the information signal has a bit width of T_b,

which is equivalent to a data rate of $1/T_b$. In that case, the bandwidth of the signal, depending on encoding technique, is roughly $2/T_b$. Similarly, the bandwidth of the pseudorandom signal is $2/T_c$, where T_c is the bit width of the pseudorandom input. The bandwidth of the combined signal is approximately the sum of the two bandwidths, or $2/(T_b + T_c)$. The amount of spreading that is achieved is a direct result of the data rate of the pseudorandom stream; the greater the data rate of the pseudorandom input, the greater the amount of spreading.

4.6 RECOMMENDED READING

It is difficult, for some reason, to find solid treatments of digital-to-digital encoding schemes. [PEEB87] provides one of the best analyses. [SKLA88] and [BENE87] also provide some insights. On the other hand, there are many good references on analog modulation schemes for digital data. Good choices are [COUC95], [HAYK94], and [PROA94]; these three also provide comprehensive treatment of digital and analog modulation schemes for analog data.

An exceptionally clear exposition that covers digital-to-analog, analog-to-digital, and analog-to-analog techniques is [PEAR92].

Both [PETE95] and [DIXO94] provide comprehensive treatment of spread spectrum.

BENE87 Benedetto, S., Biglieri, E., and Castellani, V. *Digital Transmission Theory*. Englewood Cliffs, NJ: Prentice Hall, 1987.

COUC95 Couch, L. *Modern Communication Systems: Principles and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1995.

DIXO94 Dixon, R. *Spread Spectrum Systems with Commercial Applications*. New York: Wiley, 1994.

HAYK94 Haykin, S. *Communication Systems*. New York: Wiley, 1994.

PEAR92 Pearson, J. *Basic Communication Theory*. Englewood Cliffs, NJ: Prentice Hall, 1992.

PEEB87 Peebles, P. *Digital Communication Systems*. Englewood Cliffs, NJ: Prentice Hall, 1987.

PETE95 Peterson, R., Ziemer, R., and Borth, D. *Introduction to Spread Spectrum Communications*. Englewood Cliffs, NJ: Prentice Hall, 1995.

PROA94 Proakis, J., and Salehi, M. *Communication Systems Engineering*. Englewood Cliffs, NJ: Prentice Hall, 1994.

SKLA88 Sklar, B. *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1988.

4.7 PROBLEMS

4.1 Which of the signals of Table 4.2 use differential encoding?

4.2 Develop algorithms for generating each of the codes of Table 4.2 from NRZ-L.

4.3 A modified NRZ code known as enhanced-NRZ (E-NRZ) is sometimes used for high density magnetic tape recording. E-NRZ encoding entails separating the NRZ-L data stream into 7-bit words; inverting bits 2, 3, 6, and 7; and adding one parity bit to each word. The parity bit is chosen to make the total number of 1s in the 8-bit word an odd count. What are the advantages of E-NRZ over NRZ-L? Any disadvantages?

4.4 Develop a state diagram (finite-state machine) representation of pseudoternary coding.

4.5 Consider the following signal encoding technique. Binary data are presented as input, a_m, for $m = 1, 2, 3, \ldots$ Two levels of processing occur. First, a new set of binary numbers are produced:

$$b_m = (a_m + b_{m-1}) \bmod 2$$

These are then encoded as

$$c_m = b_m - b_{m-1}$$

On reception, the original data is recovered by

$$a_m = c_m \bmod 2$$

a. Verify that the received values of a_m equal the transmitted values of a_m.
b. What sort of encoding is this?

4.6 For the bit stream 01001110, sketch the waveforms for each of the codes of Table 4.2.

4.7 The waveform of Figure 4.25 belongs to a Manchester encoded binary data stream. Determine the beginning and end of bit periods (i.e., extract clock information) and give the data sequence.

FIGURE 4.25 A Manchester stream.

4.8 A sine wave is to be used for two different signaling schemes: (a) PSK; (b) QPSK. The duration of a signal element is 10^{-5} sec. If the received signal is of the following form,

$$s(t) = 0.005 \sin (2\pi 10^6 t + \theta) \text{ volts}$$

and if the measured noise power at the receiver is 2.5×10^{-8} watts/Hz, determine the E_b/N_0 (in dB) for each case.

4.9 Consider Figure 4.9. Eight of the phases use only a single level of amplitude. The system shown encodes only 4 bits. How many bits could be encoded if the single amplitude phase were made to be double amplitude?

4.10 Derive an expression for baud rate D as a function of bit rate R for QPSK using the digital encoding techniques of Table 4.2.

4.11 What S/N ratio is required to achieve a bandwidth efficiency of 5.0 for ASK, FSK, PSK, and QPSK? Assume that the required bit error rate is 10^{-6}.

4.12 An NRZ-L signal is passed through a filter with $r = 0.5$ and then modulated onto a carrier. The data rate is 2400 bps. Evaluate the bandwidth for ASK and FSK. For FSK assume that the two frequencies used are 50 kHz and 55 kHz.

4.13 Assume that a telephone-line channel is equalized to allow bandpass data transmission over a frequency range of 600 to 3000 Hz. The available bandwidth is 2400 Hz with a center frequency of 1800 Hz. For $r = 1$, evaluate the required bandwidth for 2400 bps QPSK and 4800-bps, eight-level multilevel signaling. Is the bandwidth adequate?

4.14 Why should PCM be preferable to DM for encoding analog signals that represent digital data?

4.15 Are the modem and the codec functional inverses (i.e., could an inverted modem function as a codec, or vice versa)?

4.16 The signal of Problem 2.15 is quantized using 10-bit PCM. Find the signal-to-quantization noise ratio.

4.17 Consider an audio signal with spectral components in the range 300 to 3000 Hz. Assume that a sampling rate of 7 kHz will be used to generate a PCM signal.
a. For S/N = 30 dB, what is the number of uniform quantization levels needed? Assume $a = 0.1$
b. What data rate is required?

4.18 Find the step size δ required to prevent slope-overload noise as a function of the frequency of the highest-frequency component of the signal. Assume that all components have amplitude A.

4.19 A PCM encoder accepts a signal with a full-scale voltage of 10 V and generates 8-bit codes using uniform quantization. The maximum normalized quantized voltage is $1 - 2^{-8}$. Determine: (a) normalized step size, (b) actual step size in volts, (c) actual maximum quantized level in volts, (d) normalized resolution, (e) actual resolution, and (f) percentage resolution.

4.20 The analog waveform shown in Figure 4.26 is to be delta-modulated. The sampling period and the step size are indicated by the grid on the figure. The first DM output and the staircase function for this period are also shown. Show the rest of the staircase function and give the DM output. Indicate regions where slope-overload distortion exists.

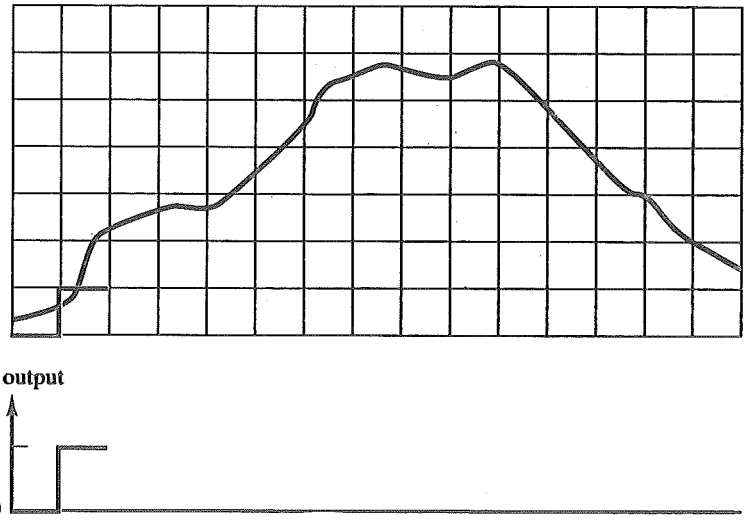


FIGURE 4.26 Delta modulation example.

4.21 By far, the most widely used technique for pseudorandom number generation is the linear congruential method. The algorithm is parameterized with four numbers, as follows:

m	the modulus	$m > 0$
a	the multiplier	$0 \leq a < m$
c	the increment	$0 \leq c < m$
X_0	the starting value, or seed	$0 \leq X_0 < m$

The sequence of pseudorandom numbers $\{X_n\}$ is obtained via the following iterative equation:

$$X_{n+1} = (aX_n + c) \bmod m$$

If m, a, c, and X_0 are integers, then this technique will produce a sequence of integers with each integer in the range $0 \leq X_n < m$. An essential characteristic of a pseudorandom number generator is that the generated sequence should appear random. Although the sequence is not random, because it is generated deterministically, there is a variety of statistical tests that can be used to assess the degree to which a sequence exhibits randomness. Another desirable characteristic is that the function should be a

full-period generating function. That is, the function should generate all the numbers between 0 and m before repeating.

With the linear congruential algorithm, a choice of parameters that provides a full period does not necessarily provide a good randomization. For example, consider the two generators

$$X_{n+1} = (6X_n) \bmod 13$$
$$X_{n+1} = (7X_n) \bmod 13$$

Write out the two sequences to show that both are full-period. Which one appears more random to you?

4.22 We would like m to be very large, so that there is the potential for producing a long series of distinct random numbers. A common criterion is that m be nearly equal to the maximum representable nonnegative integer for a given computer. Thus, a value of m near to or equal to 2^{31} is typically chosen. Many experts recommend a value of $2^{31} - 1$. You may wonder why one should not simply use 2^{31}, as this latter number can be represented with no additional bits, and the mod operation should be easier to perform. In general, the modulus $2^k - 1$ is preferable to 2^k. Why is this so?

4.23 In any use of pseudorandom numbers, whether for encryption, simulation, or statistical design, it is dangerous to blindly trust the random number generator that happens to be available in your computer's system library. One recent study found that many contemporary textbooks and programming packages make use of flawed algorithms for pseudorandom number generation. This exercise will enable you to test your system:

The test is based on a theorem attributed to Ernesto Cesaro, which states that the probability is equal to $6/\pi^2$ that the greatest common divisor of two randomly chosen integers is 1. Use this theorem in a program to determine statistically the value of π. The main program should call three subprograms: the random-number generator from the system library to generate the random integers; a subprogram to calculate the greatest common divisor of two integers using Euclid's Algorithm (found in all books on number theory), and a subprogram that calculates square roots. If these latter two programs are not available, you will have to write them as well. The main program should loop through a large number of random numbers to give an estimate of the probability referenced above. From this, it is a simple matter to solve for your estimate of π.

If the result is close to 3.14, congratulations! If not, then the result is probably low, usually a value of around 2.7. Why would such an inferior result be obtained?

4A APPENDIX

PROOF OF THE SAMPLING THEOREM

THE SAMPLING THEOREM can be restated as follows. Given that

- $x(t)$ is a bandlimited signal with bandwidth f_h.
- $p(t)$ is a sampling signal consisting of pulses at intervals $T_s = 1/f_s$, where f_s is the sampling frequency.
- $x_s(t) = x(t)p(t)$ is the sampled signal.

Then, $x(t)$ can be recovered exactly from $x_s(t)$ if and only if $f_s \geq 2f_h$.

Proof:

Because $p(t)$ consists of a uniform series of pulses, it is a periodic signal and can be represented by a Fourier series:

$$p(t) = \sum_{n=-\infty}^{\infty} P_n e^{j2\pi n f_s t}$$

We have

$$\begin{aligned} x_s(t) &= x(t)p(t) \\ &= \sum_{n=-\infty}^{\infty} P_n x(t) e^{j2\pi n f_s t} \end{aligned}$$

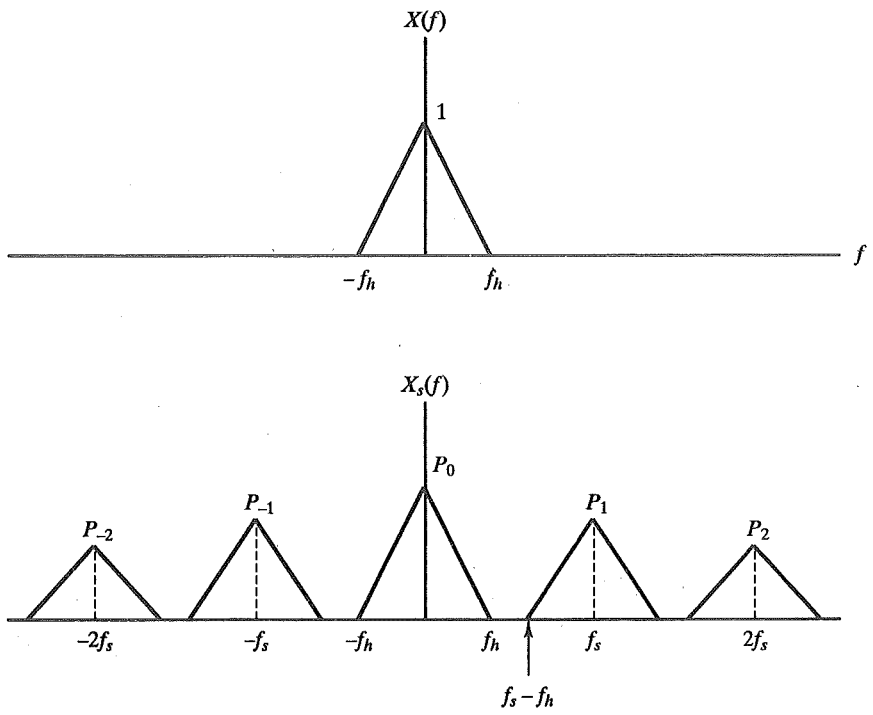


FIGURE 4.27 Spectrum of a sampled signal.

Now consider the Fourier transform of $x_s(t)$:

$$X_s(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

Substituting for x(t), we have

$$X_s(f) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} P_n x(t)e^{j2\pi nf_s t}dt$$

Rearranging yields

$$X_s(f) = \sum_{n=-\infty}^{\infty} P_n \int_{-\infty}^{\infty} x(t)e^{-j2\pi(f-nf_s)t}dt$$

From the definition of the Fourier transform, we can write

$$X_s(f - nf_s) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi(f-nf_s)t}dt$$

where $X(f)$ is the Fourier transform of $x(t)$; substituting this into the preceding equation, we have

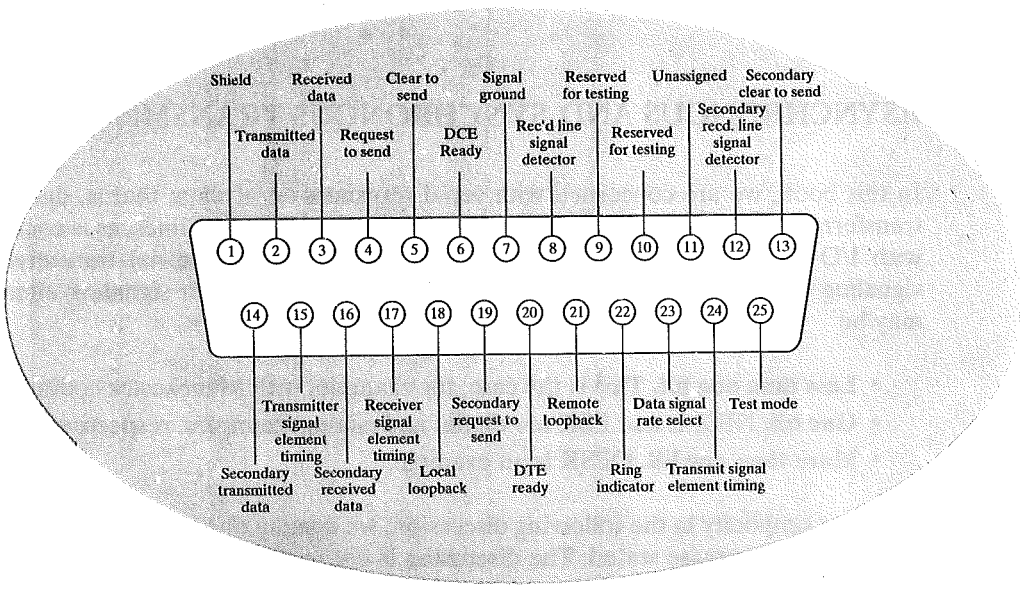
$$X_s(f) = \sum_{n=-\infty}^{\infty} P_n X(f - nf_s)$$

This last equation has an interesting interpretation, which is illustrated in Figure 4.27, where we assume without loss of generality that the bandwidth of $x(t)$ is in the range 0 to f_h. The spectrum of $x_s(t)$ is composed of the spectrum of $x(t)$ plus the spectrum of $x(t)$ translated to each harmonic of the sampling frequency. Each of the translated spectra is multiplied by the corresponding coefficient of the Fourier series of $p(t)$. Now, if $f_s > 2f_h$, these various translations do not overlap, and the spectrum of $x(t)$, multiplied by P_0, appears in $X_s(f)$. By passing $X_s(f)$ through a bandpass filter with $f < f_s$, the spectrum of $x(t)$ is recovered. In equation form,

$$X_s(f) = P_0 X(f) \qquad -fs < f < fs$$

CHAPTER 5

THE DATA COMMUNICATIONS INTERFACE



In the preceding chapters, we have been concerned primarily with the attributes of data transmission, such as the characteristics of data signals and transmission media, the encoding of signals, and transmission performance. In this chapter, we shift our emphasis to the interface between data communicating devices and the data transmission system.

For two devices linked by a transmission medium to exchange data, a high degree of cooperation is required. Typically, data are transmitted one bit at a time over the medium. The timing (rate, duration, spacing) of these bits must be the same for transmitter and receiver. Two common techniques for controlling this timing—asynchronous and synchronous—are explored in Section 5.1. Next, we look at the physical interface between data transmitting devices and the transmission line. Typically, digital data devices do not attach to and signal across the medium directly. Instead, this process is mediated through a standardized interface that provides considerable control over the interaction between the transmitting/receiving devices and the transmission line.

5.1 ASYNCHRONOUS AND SYNCHRONOUS TRANSMISSION

In this book, we are concerned with serial transmission of data; that is, data rate transferred over a single signal path rather than a parallel set of lines, as is common with I/O devices and internal computer signal paths. With serial transmission, signaling elements are sent down the line one at a time. Each signaling element may be

- **Less than one bit.** This is the case, for example, with Manchester coding.
- **One bit.** NRZ-L and FSK are digital and analog examples, respectively.
- **More than one bit.** QPSK is an example.

For simplicity in the following discussion, we assume one bit per signaling element unless otherwise stated. The discussion is not materially affected by this simplification.

Recall from Figure 2.25 that the reception of digital data involves sampling the incoming signal once per bit time to determine the binary value. One of the difficulties encountered in such a process is that various transmission impairments will corrupt the signal so that occasional errors will occur. This problem is compounded by a timing difficulty: In order for the receiver to sample the incoming bits properly, it must know the arrival time and duration of each bit that it receives.

Suppose that the sender simply transmits a stream of data bits. The sender has a clock that governs the timing of the transmitted bits. For example, if data are to be transmitted at one million bits per second (1 Mbps), then one bit will be transmitted every $1/10^6 =$ 1 microsecond (μs), as measured by the sender's clock. Typically, the receiver will attempt to sample the medium at the center of each bit-time. The receiver will time its samples at intervals of one bit-time. In our example, the sampling would occur once every 1 μs. If the receiver times its samples based on its

own clock, then there will be a problem if the transmitter's and receiver's clocks are not precisely aligned. If there is a drift of 1 percent (the receiver's clock is 1 percent faster or slower than the transmitter's clock), then the first sampling will be 0.01 of a bit-time (0.01 μs) away from the center of the bit (center of bit is .5 μs from beginning and end of bit). After 50 or more samples, the receiver may be in error because it is sampling in the wrong bit-time (50 × .01 = .5 μs). For smaller timing differences, the error would occur later, but, eventually, the receiver will be out of step with the transmitter if the transmitter sends a sufficiently long stream of bits and if no steps are taken to synchronize the transmitter and receiver.

Asynchronous Transmission

Two approaches are common for achieving the desired synchronization. The first is called, oddly enough, asynchronous transmission. The strategy with this scheme is to avoid the timing problem by not sending long, uninterrupted streams of bits. Instead, data are transmitted one character at a time, where each character is five to eight bits in length.[1] Timing or synchronization must only be maintained within each character; the receiver has the opportunity to resynchronize at the beginning of each new character.

The technique is easily explained with reference to Figure 5.1. When no character is being transmitted, the line between transmitter and receiver is in an *idle* state. The definition of idle is equivalent to the signaling element for binary 1. Thus, for NRZ-L signaling (see Figure 4.2), which is common for asynchronous transmission, idle would be the presence of a negative voltage on the line. The beginning of a character is signaled by a *start-bit* with a value of binary 0. This is followed by the five to eight bits that actually make up the character. The bits of the character are transmitted beginning with the least significant bit. For example, for ASCII characters, the first bit transmitted is the bit labeled b_1 in Table 2.1. Usually, this is followed by a parity bit, which therefore is in the most significant bit position. The parity bit is set by the transmitter such that the total number of ones in the character, including the parity bit, is even (even parity) or odd (odd parity), depending on the convention being used. This bit is used by the receiver for error detection, as discussed in Chapter 6. The final element is a *stop*, which is a binary 1. A minimum length for the stop is specified, and this is usually 1, 1.5, or 2 times the duration of an ordinary bit. No maximum value is specified. Because the stop is the same as the idle state, the transmitter will continue to transmit the stop signal until it is ready to send the next character.

If a steady stream of characters is sent, the interval between two characters is uniform and equal to the stop element. For example, if the stop is one bit-time and the ASCII characters ABC are sent (with even parity bit), the pattern is 0100000101001000010101100001111 . . . 111.[2] The start bit (0) starts the timing sequence for the next nine elements, which are the 8-bit ASCII code and the stop

[1] The number of bits that comprise a character depends on the code used. We have already seen one common example, the ASCII code, which uses seven bits per character (Table 4-1). Another common code is the Extended Binary Coded Decimal Interchange Code (EBCDIC), which is an 8-bit character code used on all IBM machines except for their personal computers.

[2] In the text, the transmission is shown from left (first bit transmitted) to right (last bit transmitted).

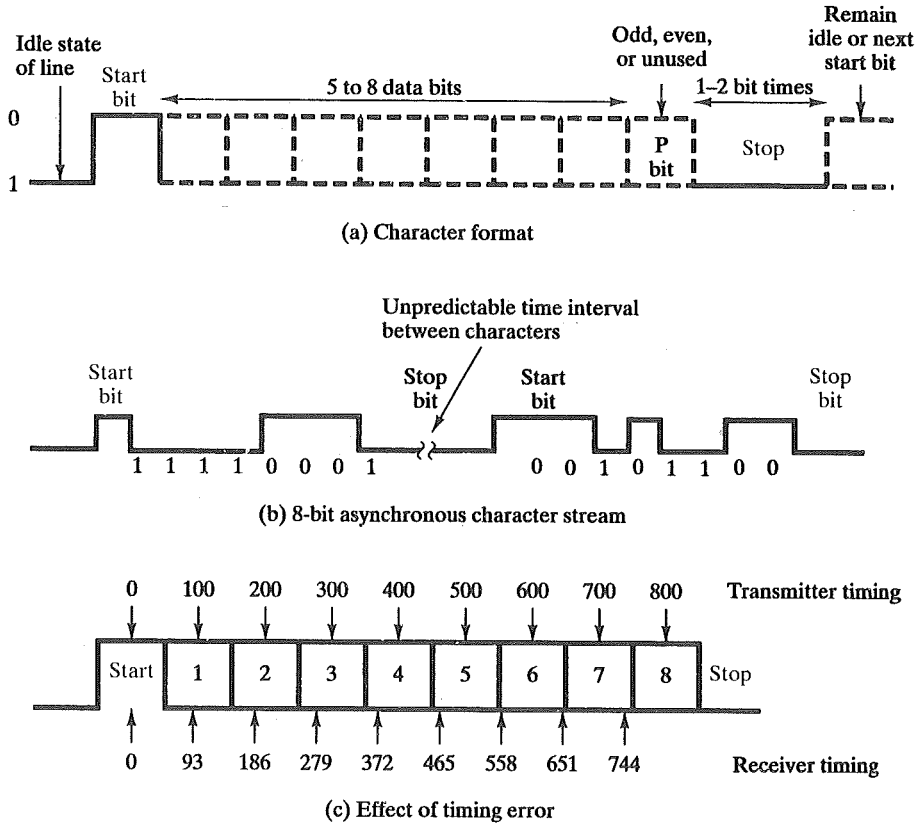


FIGURE 5.1 Asynchronous transmission.

bit. In the idle state, the receiver looks for a transition from 1 to 0 to signal the beginning of the next character and then samples the input signal at one-bit intervals for seven intervals. It then looks for the next 1-to-0 transition, which will occur no sooner than one more bit-time.

The timing requirements for this scheme are modest. For example, ASCII characters are typically sent as 8-bit units, including the parity bit. If the receiver is 5 percent slower or faster than the transmitter, the sampling of the eighth information bit will be displaced by 45 percent and still be correctly sampled. Figure 5.1c shows the effects of a timing error of sufficient magnitude to cause an error in reception. In this example we assume a data rate of 10,000 bits per second (10 kbps); therefore, each bit is of 0.1 millisecond (ms), or 100 μs, duration. Assume that the receiver is off by 7 percent, or 7 μs per bit-time. Thus, the receiver samples the incoming character every 93 μs (based on the transmitter's clock). As can be seen, the last sample is erroneous.

An error such as this actually results in two errors. First, the last sampled bit is incorrectly received. Second, the bit count may now be out of alignment. If bit 7 is a 1 and bit 8 is a 0, bit 8 could be mistaken for a start bit. This condition is termed

a *framing error*, as the character plus start and stop bits are sometimes referred to as a frame. A framing error can also occur if some noise condition causes the false appearance of a start bit during the idle state.

Asynchronous transmission is simple and cheap but requires an overhead of two to three bits per character. For example, for an 8-bit code, using a 1-bit-long stop bit, two out of every ten bits convey no information but are there merely for synchronization; thus the overhead is 20%. Of course, the percentage overhead could be reduced by sending larger blocks of bits between the start and stop bits. However, as Figure 5.1c indicates, the larger the block of bits, the greater the cumulative timing error. To achieve greater efficiency, a different form of synchronization, known as synchronous transmission, is used.

Synchronous Transmission

With synchronous transmission, a block of bits is transmitted in a steady stream without start and stop codes. The block may be many bits in length. To prevent timing drift between transmitter and receiver, their clocks must somehow be synchronized. One possibility is to provide a separate clock line between transmitter and receiver. One side (transmitter or receiver) pulses the line regularly with one short pulse per bit-time. The other side uses these regular pulses as a clock. This technique works well over short distances, but over longer distances the clock pulses are subject to the same impairments as the data signal, and timing errors can occur. The other alternative is to embed the clocking information in the data signal; for digital signals, this can be accomplished with Manchester or Differential Manchester encoding. For analog signals, a number of techniques can be used; for example, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier.

With synchronous transmission, there is another level of synchronization required, so as to allow the receiver to determine the beginning and end of a block of data; to achieve this, each block begins with a *preamble* bit pattern and generally ends with a *postamble* bit pattern. In addition, other bits are added to the block that convey control information used in the data link control procedures discussed in Chapter 6. The data plus preamble, postamble, and control information are called a **frame**. The exact format of the frame depends on which data link control procedure is being used.

Figure 5.2 shows, in general terms, a typical frame format for synchronous transmission. Typically, the frame starts with a preamble called a flag, which is eight bit-long. The same flag is used as a postamble. The receiver looks for the occurrence of the flag pattern to signal the start of a frame. This is followed by some number of control fields, then a data field (variable length for most protocols), more control fields, and finally the flag is repeated.

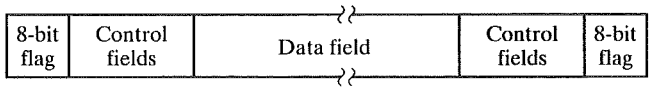


FIGURE 5.2 Synchronous frame format.

For sizable blocks of data, synchronous transmission is far more efficient than asynchronous. Asynchronous transmission requires 20 percent or more overhead. The control information, preamble, and postamble in synchronous transmission are typically less than 100 bits. For example, one of the more common schemes, HDLC, contains 48 bits of control, preamble, and postamble. Thus, for a 1000-character block of data, each frame consists of 48 bits of overhead and $1000 \times 8 = 8{,}000$ bits of data, for a percentage overhead of only $48/8048 \times 100\% = 0.6\%$.

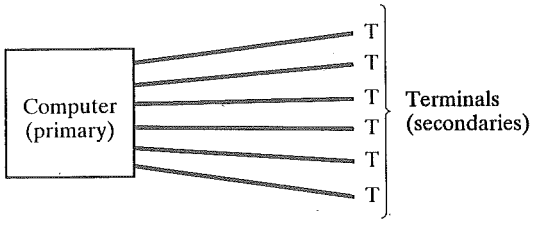
5.2 LINE CONFIGURATIONS

Two characteristics that distinguish various data link configurations are topology and whether the link is half duplex or full duplex.

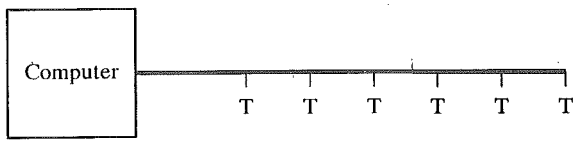
Topology

The topology of a data link refers to the physical arrangement of stations on a transmission medium. If there are only two stations, (e.g., a terminal and a computer or two computers), the link is point-to-point. If there are more than two stations, then it is a multipoint topology. Traditionally, a multipoint link has been used in the case of a computer (primary station) and a set of terminals (secondary stations). In today's environments, the multipoint topology is found in local area networks.

Traditional multipoint topologies are made possible when the terminals are only transmitting a fraction of the time. Figure 5.3 illustrates the advantages of the multipoint configuration. If each terminal has a point-to-point link to its computer, then the computer must have one I/O port for each terminal. Also, there is a sepa-



(a) Point-to-point



(b) Multipoint

FIGURE 5.3 Traditional computer/terminal configurations.

rate transmission line from the computer to each terminal. In a multipoint configuration, the computer needs only a single I/O port, thereby saving hardware costs. Only a single transmission line is needed, which also saves costs.

Full Duplex and Half Duplex

Data exchanges over a transmission line can be classified as full duplex or half duplex. With *half-duplex transmission*, only one of two stations on a point-to-point link may transmit at a time. This mode is also referred to as *two-way alternate*, suggestive of the fact that two stations must alternate in transmitting; this can be compared to a one-lane, two-way bridge. This form of transmission is often used for terminal-to-computer interaction. While a user is entering and transmitting data, the computer is prevented from sending data, which would appear on the terminal screen and cause confusion.

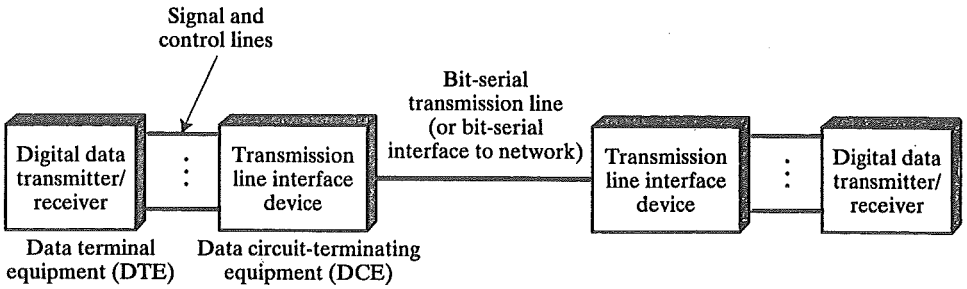
For *full-duplex transmission*, two stations can simultaneously send and receive data from each other. Thus, this mode is known as *two-way simultaneous* and may be compared to a two-lane, two-way bridge. For computer-to-computer data exchange, this form of transmission is more efficient than half-duplex transmission.

With digital signaling, which requires guided transmission, full-duplex operation usually requires two separate transmission paths (e.g., two twisted pairs), while half duplex requires only one. For analog signaling, it depends on frequency; if a station transmits and receives on the same frequency, it must operate in half-duplex mode for wireless transmission, although it may operate in full-duplex mode for guided transmission using two separate transmission lines. If a station transmits on one frequency and receives on another, it may operate in full-duplex mode for wireless transmission and in full-duplex mode with a single line for guided transmission.

5.3 INTERFACING

Most digital data-processing devices have limited data-transmission capability. Typically, they generate a simple digital signal, such as NRZ-L, and the distance across which they can transmit data is limited. Consequently, it is rare for such a device (terminal, computer) to attach directly to a transmission or networking facility. The more common situation is depicted in Figure 5.4. The devices we are discussing, which include terminals and computers, are generically referred to as *data terminal equipment (DTE)*. A DTE makes use of the transmission system through the mediation of *data circuit-terminating equipment (DCE)*. An example of the latter is a modem.

On one side, the DCE is responsible for transmitting and receiving bits, one at a time, over a transmission medium or network. On the other side, the DCE must interact with the DTE. In general, this requires both data and control information to be exchanged. This is done over a set of wires referred to as *interchange circuits*. For this scheme to work, a high degree of cooperation is required. The two DCEs that exchange signals over the transmission line or network must understand each



(a) Generic interface to transmission medium

(b) Typical configuration

FIGURE 5.4 Data communications interfacing.

other. That is, the receiver of each must use the same encoding scheme (e.g., Manchester, PSK) and data rate as the transmitter of the other. In addition, each DTE-DCE pair must be designed to interact cooperatively. To ease the burden on data-processing equipment manufacturers and users, standards have been developed that specify the exact nature of the interface between the DTE and the DCE. Such an interface has four important characteristics:

- Mechanical
- Electrical
- Functional
- Procedural

The *mechanical characteristics* pertain to the actual physical connection of the DTE to the DCE. Typically, the signal and control interchange circuits are bundled into a cable with a terminator plug, male or female, at each end. The DTE and DCE must present plugs of opposite genders at one end of the cable, effecting the physical connection; this is analogous to the way residential electrical power is produced. Power is provided via a socket or wall outlet, and the device to be attached must have the appropriate male plug (two-pronged, two-pronged polarized, or three-pronged) to match the socket.

The *electrical characteristics* have to do with the voltage levels and timing of voltage changes. Both DTE and DCE must use the same code (e.g., NRZ-L), must use the same voltage levels to mean the same things, and must use the same dura-

tion of signal elements. These characteristics determine the data rates and distances that can be achieved.

Functional characteristics specify the functions that are performed by assigning meanings to each of the interchange circuits. Functions can be classified into the broad categories of data, control, timing, and electrical ground.

Procedural characteristics specify the sequence of events for transmitting data, based on the functional characteristics of the interface. The examples that follow should clarify this point.

A variety of standards for interfacing exists; this section presents two of the most important: V.24/EIA-232-E, and the ISDN Physical Interface.

V.24/EIA-232-E

The most widely used interface is one that is specified in the ITU-T standard, V.24. In fact, this standard specifies only the functional and procedural aspects of the interface; V.24 references other standards for the electrical and mechanical aspects. In the United States, there is a corresponding specification, virtually identical, that covers all four aspects: EIA-232. The correspondence is as follows:

- Mechanical: ISO 2110
- Electrical: V.28
- Functional: V.24
- Procedural: V.24

EIA-232 was first issued by the Electronic Industries Association in 1962, as RS-232. It is currently in its fifth revision EIA-232-E, issued in 1991. The current V.24 and V.28 specifications were issued in 1993. This interface is used to connect DTE devices to voice-grade modems for use on public analog telecommunications systems. It is also widely used for many other interconnection applications.

Mechanical Specification

The mechanical specification for EIA-232-E is illustrated in Figure 5.5. It calls for a 25-pin connector, defined in ISO 2110, with a specific arrangement of leads. This connector is the terminating plug or socket on a cable running from a DTE (e.g., terminal) or DCE (e.g., modem). Thus, in theory, a 25-wire cable could be used to connect the DTE to the DCE. In practice, far fewer interchange circuits are used in most applications.

Electrical Specification

The electrical specification defines the signaling between DTE and DCE. Digital signaling is used on all interchange circuits. Depending on the function of the interchange circuit, the electrical values are interpreted either as binary or as control signals. The convention specifies that, with respect to a common ground, a voltage more negative than –3 volts is interpreted as binary 1 and a voltage more positive than +3 volts is interpreted as binary 0; this is the NRZ-L code illustrated in Figure 4.2. The interface is rated at a signal rate of <20 kbps and a distance of <15 meters.

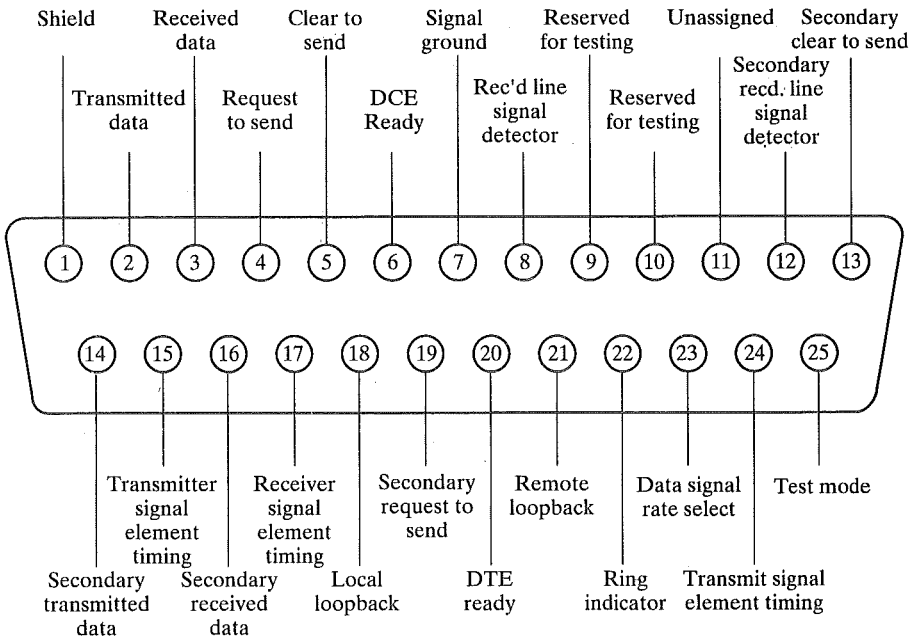


FIGURE 5.5 Pin assignments for V.24/EIA-232.

Greater distances and data rates are possible with good design, but it is prudent to assume that these limits apply in practice as well as in theory.

The same voltage levels apply to control signals; a voltage more negative than –3 volts is interpreted as an OFF condition and a voltage more positive than +3 volts is interpreted as an ON condition.

Functional Specification

Table 5.1 summarizes the functional specification of the interchange circuits, and Figure 5.5 illustrates the placement of these circuits on the plug. The circuits can be grouped into the categories of data, control, timing, and ground. There is one data circuit in each direction, so full-duplex operation is possible. In addition, there are two secondary data circuits that are useful when the device operates in a half-duplex fashion. In the case of half-duplex operation, data exchange between two DTEs (via their DCEs and the intervening communications link) is only conducted in one direction at a time. However, there may be a need to send a halt or flow-control message to a transmitting device; to accommodate this, the communication link is equipped with a reverse channel, usually at a much lower data rate than the primary channel. At the DTE-DCE interface, the reverse channel is carried on a separate pair of data circuits.

There are fifteen control circuits. The first ten of these listed in Table 5.1 relate to the transmission of data over the primary channel. For asynchronous transmission, six of these circuits are used (105, 106, 107, 108.2, 125, 109). The use of these circuits is explained in the subsection on procedural specifications. In addition to these six circuits, three other control circuits are used in synchronous transmis-

TABLE 5.1 V.24/EIA-232-E interchange circuits.

V.24	EIA-232	Name	Direction to:	Function
		DATA SIGNALS		
103	BA	Transmitted data	DCE	Transmitted by DTE
104	BB	Received data	DTE	Received by DTE
118	SBA	Secondary transmitted data	DCE	Transmitted by DTE
104	SBB	Secondary received data	DTE	Received by DTE
		CONTROL SIGNALS		
105	CA	Request to send	DCE	DTE wishes to transmit
106	CB	Clear to send	DTE	DCE is ready to receive; response to Request to send
107	CC	DCE ready	DTE	DCE is ready to operate
108.2	CD	DTE ready	DCE	DTE is ready to operate
125	CE	Ring indicator	DTE	DCE is receiving a ringing signal on the channel line
109	CF	Received line signal detector	DTE	DCE is receiving a signal within appropriate limits on the channel line
110	CG	Signal quality detector	DTE	Indicates whether there is a high probability of error in the data received
111	CH	Data signal rate selector	DCE	Selects one of two data rates
112	CI	Data signal rate selector	DTE	Selects one of two data rates
133	CJ	Ready for receiving	DCE	On/off flow control
120	SCA	Secondary request to send	DCE	DTE wishes to transmit on reverse channel
121	SCB	Secondary clear to send	DTE	DCE is ready to receive on reverse channel
122	SCF	Secondary received line signal detector	DTE	Same as 109, for reverse channel
140	RL	Remote loopback	DCE	Instructs remote DCE to loop back signals
141	LL	Local loopback	DCE	Instructs DCE to loop back signals
142	TM	Test mode	DTE	Local DCE is in a test condition
		TIMING SIGNALS		
113	DA	Transmitter signal element timing	DCE	Clocking signal; transitions to ON and OFF occur at center of each signal element
114	DB	Transmitter signal element timing	DTE	Clocking signal; both 113 and 114 relate to signals on circuit 103
115	DD	Receiver signal element timing	DTE	Clocking signal for circuit 104
		GROUND		
102	AB	Signal ground/common return		Common ground reference for all circuits

sion. The Signal Quality Detector circuit is turned ON by the DCE to indicate that the quality of the incoming signal over the telephone line has deteriorated beyond some defined threshold. Most high-speed modems support more than one transmission rate so that they can fall back to a lower speed if the telephone line becomes noisy. The Data Signal Rate Selector circuits are used to change speeds; either the DTE or DCE may initiate the change. The next three control circuits (120, 121, 122) are used to control the use of the secondary channel, which may be used as a reverse channel or for some other auxiliary purpose.

The last group of control signals relate to loopback testing. These circuits allow the DTE to cause the DCE to perform a loopback test. These circuits are only valid if the modem or other DCE supports loopback control; this is now a common modem feature. In the local loopback function, the transmitter output of the modem is connected to the receiver input, disconnecting the modem from the transmission line. A stream of data generated by the user device is sent to the modem and looped back to the user device. For remote loopback, the local modem is connected to the transmission facility in the usual fashion, and the receiver output of the remote modem is connected to the modem's transmitter input. During either form of test, the DCE turns ON the Test Mode circuit. Table 5.2 shows the settings for all of the circuits related to loopback testing, and Figure 5.6 illustrates the use.

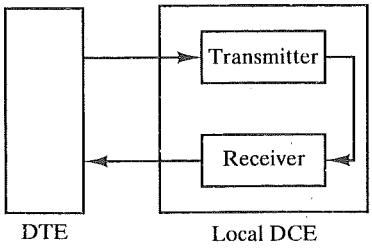
Loopback control is a useful fault-isolation tool. For example, suppose that a user at a personal computer is communicating with a server by means of a modem connection and communication suddenly ceases. The problem could be with the local modem, the communications facility, the remote modem, or the remote server. A network manager can use loopback tests to isolate the fault. Local loopback checks the functioning of the local interface and the local DCE. Remote loopback tests the operation of the transmission channel and the remote DCE.

The timing signals provide clock pulses for synchronous transmission. When the DCE is sending synchronous data over the Received Data circuit (104), it also sends 1-0 and 0-1 transitions on Receiver Element Signal Timing (115), with transitions timed to the middle of each BB signal element. When the DTE is sending synchronous data, either the DTE or DCE can provide timing pulses, depending on the circumstances.

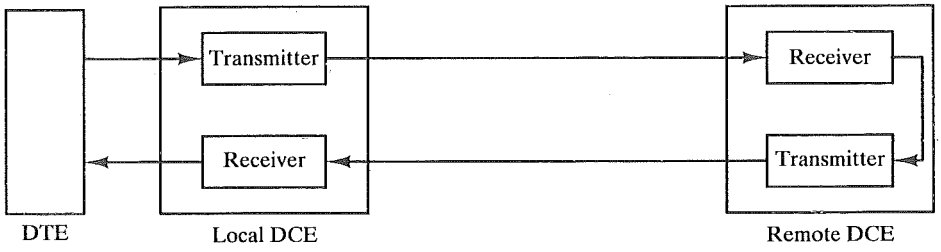
Finally, the signal ground/common return (102) serves as the return circuit for all data leads. Hence, transmission is unbalanced, with only one active wire. Balanced and unbalanced transmission are discussed in the section on the ISDN interface.

TABLE 5.2 Loopback circuit settings for V.24/EIA-232.

Local loopback		Remote loopback		
Circuit	Condition	Circuit	Local interface	Remote interface
DCE ready	ON	DCE ready	ON	OFF
Local loopback	ON	Local loopback	OFF	OFF
Remote loopback	OFF	Remote loopback	ON	OFF
Test mode	ON	Test mode	ON	ON



(a) Local loopback Testing



(b) Remote loopback testing

FIGURE 5.6 Local and remote loopback.

Procedural Specification

The procedural specification defines the sequence in which the various circuits are used for a particular application. We give a few examples.

The first example is a very common one for connecting two devices over a short distance within a building. It is known as an asynchronous private line modem, or a limited-distance modem. As the name suggests, the limited-distance modem accepts digital signals from a DTE, such as a terminal or computer, converts these to analog signals, and then transmits these over a short length of medium, such as twisted pair. On the other end of the line is another limited-distance modem, which accepts the incoming analog signals, converts them to digital, and passes them on to another terminal or computer. Of course, the exchange of data is two-way. For this simple application, only the following interchange circuits are actually required:

- Signal ground (102)
- Transmitted data (103)
- Received data (104)
- Request to send (105)
- Clear to send (106)
- DCE ready (107)
- Received-Line Signal Detector (109)

When the modem (DCE) is turned on and is ready to operate, it asserts (applies a constant negative voltage to) the DCE Ready line. When the DTE is

ready to send data (e.g., the terminal user has entered a character), it asserts Request to Send. The modem responds, when ready, by asserting Clear to Send, indicating that data may be transmitted over the Transmitted Data line. If the arrangement is half-duplex, then Request to Send also inhibits the receive mode. The DTE may now transmit data over the Transmitted Data line. When data arrive from the remote modem, the local modem asserts Received-Line Signal Detector to indicate that the remote modem is transmitting and delivers the data on the Received Data line. Note that it is not necessary to use timing circuits, as this is asynchronous transmission.

The circuits just listed are sufficient for private line point-to-point modems, but additional circuits are required to use a modem to transmit data over the telephone network. In this case, the initiator of a connection must call the destination device over the network. Two additional leads are required:

- DTE ready (108.2)
- Ring indicator (125)

With the addition of these two lines, the DTE-modem system can effectively use the telephone network in a way analogous to voice telephone usage. Figure 5.7 depicts the steps involved in dial-up half-duplex operation. When a call is made, either manually or automatically, the telephone system sends a ringing signal. A telephone set would respond by ringing its bell; a modem responds by asserting Ring Indicator. A person answers a call by lifting the handset; a DTE answers by asserting Data Terminal Ready. A person who answers a call will listen for another's voice, and, if nothing is heard, hang up. A DTE will listen for Carrier Detect, which will be asserted by the modem when a signal is present; if this circuit is not asserted, the DTE will drop Data Terminal Ready. You might wonder how this last contingency might arise; one common way is if a person accidentally dials the number of a modem. This activates the modem's DTE, but when no carrier tone comes through, the problem is resolved.

It is instructive to consider situations in which the distances between devices are so close as to allow two DTEs to directly signal each other. In this case, the V.24/EIA-232 interchange circuits can still be used, but no DCE equipment is provided. For this scheme to work, a null modem is needed, which interconnects leads in such a way as to fool both DTEs into thinking that they are connected to modems. Figure 5.8 is an example of a null modem configuration; the reasons for the particular connections should be apparent to the reader who has grasped the preceding discussion.

ISDN Physical Interface

The wide variety of functions available with V.24/EIA-232 is provided by the use of a large number of interchange circuits. This is a rather expensive way to achieve results. An alternative would be to provide fewer circuits but to add more logic at the DTE and DCE interfaces. With the dropping costs of logic circuitry, this is an attractive approach. This approach was taken in the X.21 standard for interfacing to public circuit-switched networks, specifying a 15-pin connector. More recently, the

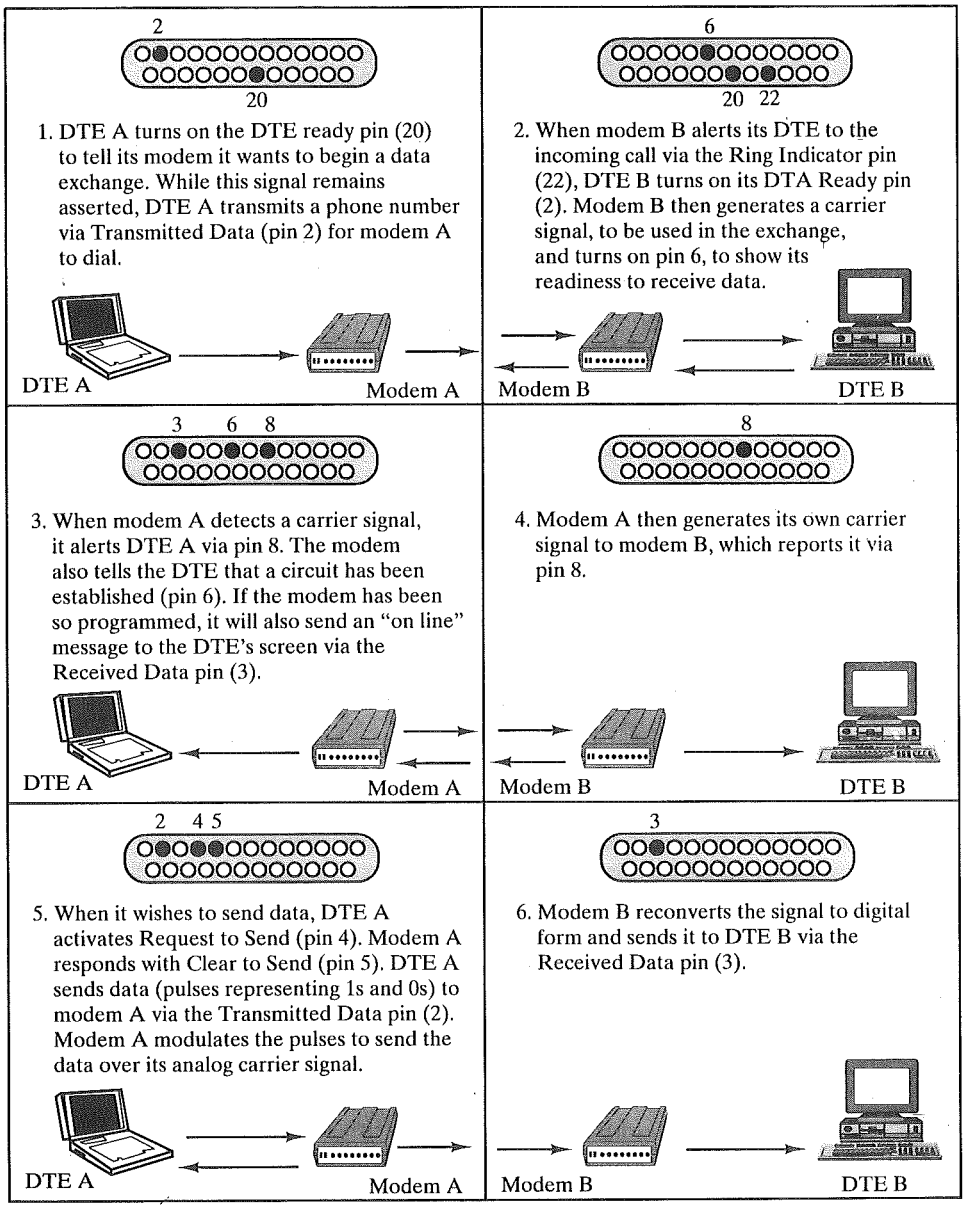


FIGURE 5.7 V.24/EIA-232 dial-up operation.

trend has been carried further with the specification of an 8-pin physical connector to an Integrated Services Digital Network (ISDN). ISDN, which is an all-digital replacement for existing public telephone and analog telecommunications networks, is discussed further in Appendix A. In this section, we look at the physical interface defined for ISDN.

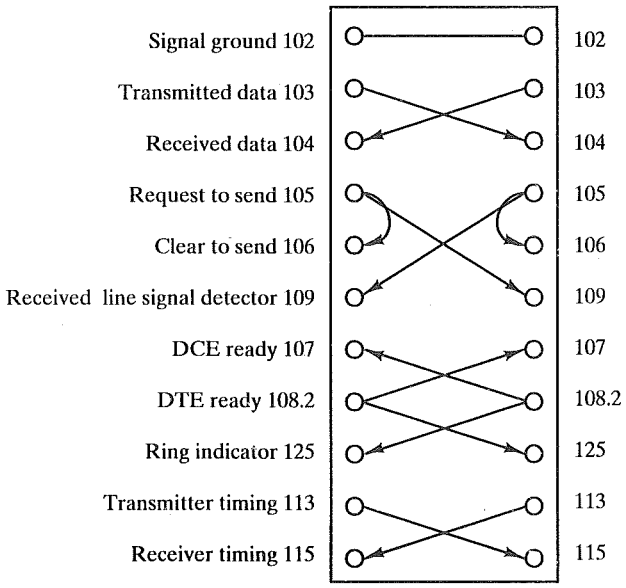


FIGURE 5.8 Example of a null modem.

Physical Connection

In ISDN terminology, a physical connection is made between terminal equipment (TE) and network-terminating equipment (NT). For purposes of our discussion, these terms correspond, rather closely, to DTE and DCE, respectively. The physical connection, defined in ISO 8877, specifies that the NT and TE cables shall terminate in matching plugs that provide for 8 contacts.

Figure 5.9 illustrates the contact assignments for each of the 8 lines on both the NT and TE sides. Two pins are used to provide data transmission in each direction. These contact points are used to connect twisted-pair leads coming from the NT and TE devices. Because there are no specific functional circuits, the transmit/receive circuits are used to carry both data and control signals. The control information is transmitted in the form of messages.

The specification provides for the capability to transfer power across the interface. The direction of power transfer depends on the application. In a typical application, it may be desirable to provide for power transfer from the network side toward the terminal in order, for example, to maintain a basic telephony service in the event of failure of the locally provided power. This power transfer can be accomplished using the same leads used for digital signal transmission (c, d, e, f), or on additional wires, using access leads g-h. The remaining two leads are not used in the ISDN configuration but may be useful in other configurations.

Electrical Specification

The ISDN electrical specification dictates the use of balanced transmission. With *balanced transmission*, signals are carried on a line, such as twisted pair, consisting

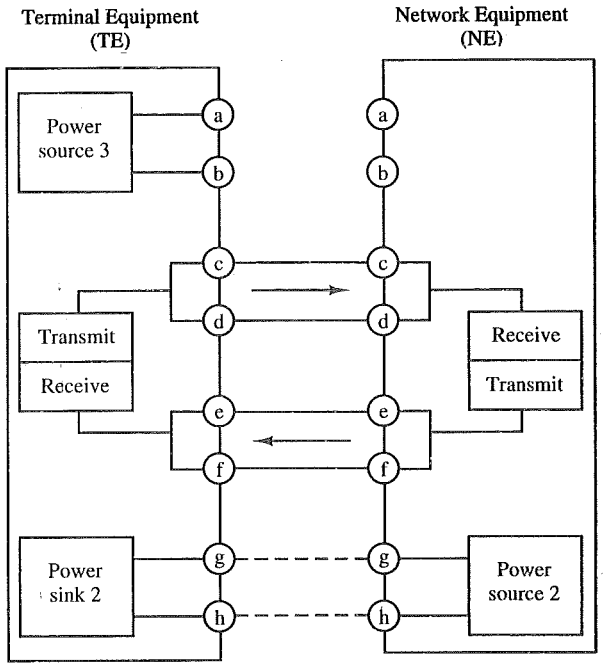


FIGURE 5.9 ISDN interface.

of two conductors. Signals are transmitted as a current that travels down one conductor and returns on the other, the two conductors forming a complete circuit. For digital signals, this technique is known as *differential signaling*,[3] as the binary value depends on the direction of the voltage difference between the two conductors. *Unbalanced transmission*, which is used on older interfaces such as EIA-232, uses a single conductor to carry the signal, with ground providing the return path.

The balance mode tolerates more, and produces less, noise than unbalanced mode. Ideally, interference on a balanced line will act equally on both conductors and not affect the voltage difference. Because unbalanced transmission does not possess these advantages, it is generally limited to use on coaxial cable; when it is used on interchange circuits, such as EIA-232, it is limited to very short distances.

The data encoding format used on the ISDN interface depends on the data rate. For the *basic rate* of 192 kbps, the standard specifies the use of pseudoternary coding (Figure 4.2). Binary one is represented by the absence of voltage, and binary zero is represented by a positive or negative pulse of 750 mV ±10%. For the *primary rate*, there are two options: 1.544 Mbps using alternate mark inversion (AMI) with B8ZS (Figure 4.6) and 2.049 Mbps using AMI with HDB3. The reason for the different schemes for the two different primary rates is simply historical; neither has a particular advantage.

[3] Not to be confused with differential encoding; see Section 4.1.

5.4 RECOMMENDED READING

[BLAC95a] provides detailed, broad coverage of many physical-layer interface standards.

[BLAC95b] focuses on the ITU-T V series recommendations. [SEYE91] is an easy-to-read and thorough introduction to EIA-232.

BLAC95a Black, U. *Physical Level Interfaces and Protocols*. Los Alamitos, CA: IEEE Computer Society Press, 1995.

BLAC95b Black, U. *The V Series Recommendations: Standards for Data Communications Over the Telephone Network*. New York: McGraw-Hill, 1995.

SEYE91 Seyer, M. *RS-232 Made Easy: Connecting Computers, Printers, Terminals, and Modems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

5.5 PROBLEMS

5.1 A data source produces 8-bit ASCII characters. Derive an expression for the maximum data rate (rate of ASCII data bits) over a B-bps line for the following:
   a. Asynchronous transmission with a 1.5-unit stop bit.
   b. Synchronous transmission, with a frame consisting of 48 control bits and 128 information bits. The information field contains 8-bit ASCII characters.
   c. Same as (b), but with an information field of 1024 bits.

5.2 Demonstrate by example (write down a few dozen arbitrary bit patterns with start and stop bits) that a receiver that suffers a framing error on asynchronous transmission will eventually become realigned.

5.3 Suppose that the sender and receiver agree not to use any stop bits. Could this work? If so, explain any necessary conditions.

5.4 Consider a transmission system that is clocked by a master clock running at 8 MHz. This clock has a maximum error of 30 seconds per month. Transmission is asynchronous serial consisting of characters containing one start bit, seven data bits, one parity bit, and one stop bit. If characters are transmitted in a continuous stream as rapidly as possible (a burst mode), how many characters could be sent before a transmission error caused by the master clock error occurs? Assume that each bit must be sampled within 40% of its center position. Note that the transmission rate is not a factor, as both the bit period and the absolute timing error decrease proportionately at higher transmission rates.

5.5 An asynchronous transmission uses 8 data bits, an even parity bit, and 2 stop bits. What percentage of clock inaccuracy can be tolerated at the receiver with respect to the framing error? Assume that the bit samples are taken at the middle of the clock period. Also assume that, at the beginning of the start bit, the clock and incoming bits are in phase.

5.6 Suppose that a synchronous serial data transmission is clocked by two 8-MHz clocks (one at the sender and one at the receiver) that each have a drift of 1 minute in one year. How long a sequence of bits can be sent before possible clock drift could cause a problem? Assume that a bit waveform will be good if it is sampled within 40% of its center and that the sender and receiver are resynchronized at the beginning of each frame.

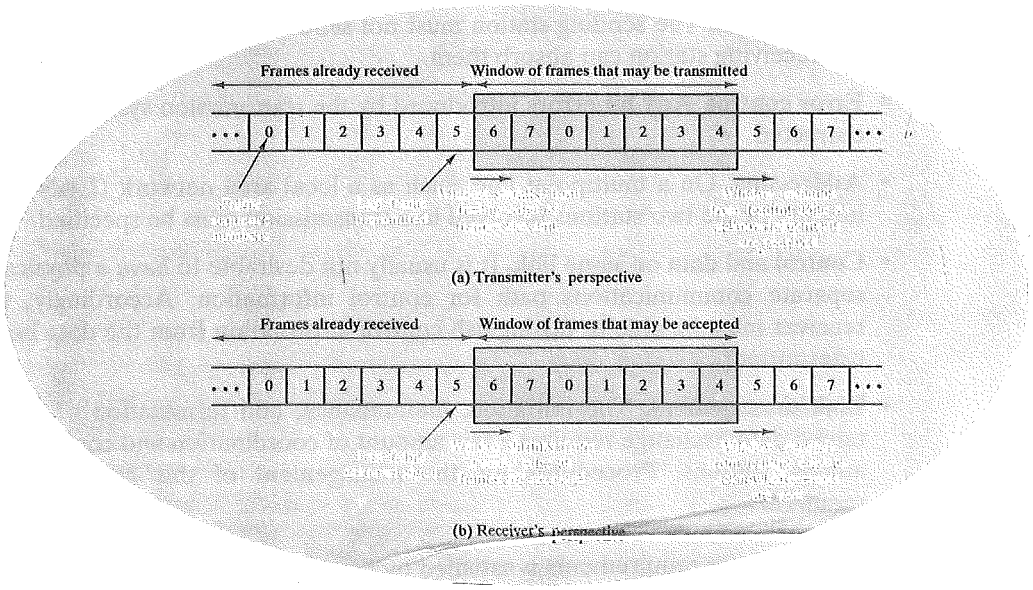
5.7 Draw a timing diagram showing the state of all EIA-232 leads between two DTE-DCE pairs during the course of a data call on the switched telephone network.

5.8 Explain the operation of each null modem connection in Figure 5.8.

5.9 For the V.24/EIA-232 Remote Loopback circuit to function properly, what circuits must be logically connected?

CHAPTER 6

DATA LINK CONTROL



Our discussion so far has concerned *sending signals over a transmission link.* For effective digital data communications, much more is needed to control and manage the exchange. In this chapter, we shift our emphasis to that of *sending data over a data communications link.* To achieve the necessary control, a layer of logic is added above the physical interfacing discussed in Chapter 5; this logic is referred to as *data link control* or a *data link control protocol.* When a data link control protocol is used, the transmission medium between systems is referred to as a *data link.*

To see the need for data link control, we list some of the requirements and objectives for effective data communication between two directly connected transmitting-receiving stations:

- **Frame synchronization.** Data are sent in blocks called frames. The beginning and end of each frame must be recognizable. We briefly introduced this topic with the discussion of synchronous frames (Figure 5.2).
- **Flow control.** The sending station must not send frames at a rate faster then the receiving station can absorb them.
- **Error control.** Any bit errors introduced by the transmission system must be corrected.
- **Addressing.** On a multipoint line, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.
- **Control and data on same link.** It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.
- **Link management.** The initiation, maintenance, and termination of a sustained data exchange requires a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.

None of these requirements is satisfied by the physical interfacing techniques described in Chapter 6. We shall see in this chapter that a data link protocol that satisfies these requirements is a rather complex affair. We begin by looking at three key mechanisms that are part of data link control: flow control, error detection, and error control. Following this background information, we look at the most important example of a data link control protocol: HDLC (high-level data link control). This protocol is important for two reasons: First, it is a widely used standardized data link control protocol. And secondly, HDLC serves as a baseline from which virtually all other important data link control protocols are derived. Following a detailed examination of HDLC, these other protocols are briefly surveyed. Finally, an appendix to this chapter addresses some performance issues relating to data link control.

6.1 FLOW CONTROL

Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data. The receiving entity typically allocates a data buffer of some maximum length for a transfer. When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software. In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data.

To begin, we examine mechanisms for flow control in the absence of errors. The model we will use is depicted in Figure 6.1a, which is a vertical-time sequence diagram. It has the advantages of showing time dependencies and illustrating the correct send-receive relationship. Each arrow represents a single frame transiting a data link between two stations. The data are sent in a sequence of frames with each frame containing a portion of the data and some control information. For now, we

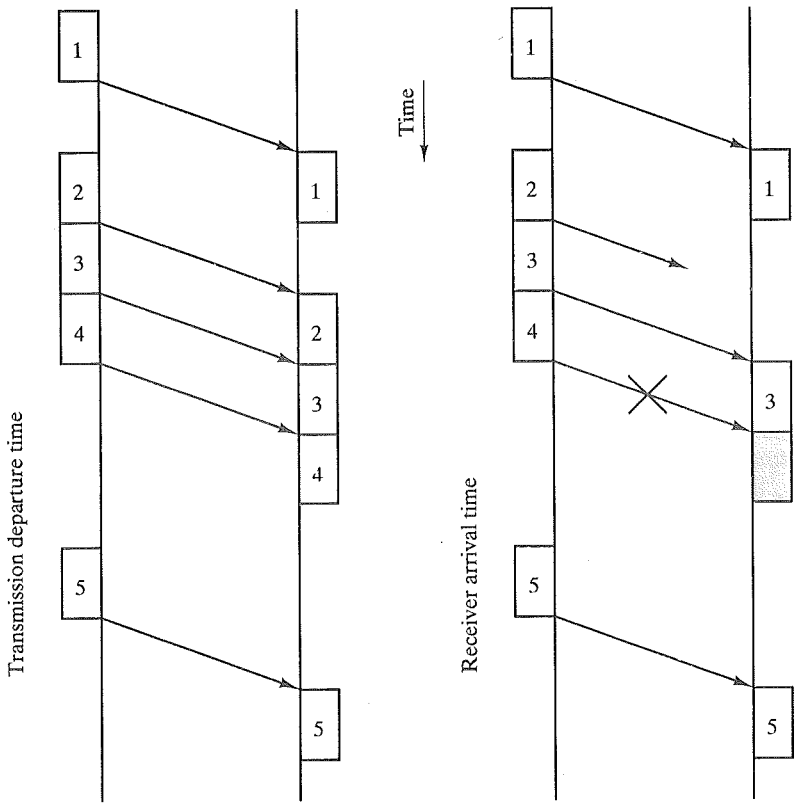


FIGURE 6.1 Model of frame transmission.

assume that all frames that are transmitted are successfully received; no frames are lost and none arrive with errors. Furthermore, frames arrive in the same order in which they are sent. However, each transmitted frame suffers an arbitrary and variable amount of delay before reception.

Stop-and-Wait Flow Control

The simplest form of flow control, known as stop-and-wait flow control, works as follows. A source entity transmits a frame. After reception, the destination entity indicates its willingness to accept another frame by sending back an acknowledgment to the frame just received. The source must wait until it receives the acknowledgment before sending the next frame. The destination can thus stop the flow of data by simply withholding acknowledgment. This procedure works fine and, indeed, can hardly be improved upon when a message is sent in a few large frames. However, it is often the case that a source will break up a large block of data into smaller blocks and transmit the data in many frames. This is done for the following reasons:

- The buffer size of the receiver may be limited.
- The longer the transmission, the more likely that there will be an error, necessitating retransmission of the entire frame. With smaller frames, errors are detected sooner, and a smaller amount of data needs to be retransmitted.
- On a shared medium, such as a LAN, it is usually desirable not to permit one station to occupy the medium for an extended period, as this causes long delays at the other sending stations.

With the use of multiple frames for a single message, the stop-and-wait procedure may be inadequate. The essence of the problem is that only one frame at a time can be in transit. In situations where the bit length of the link is greater than the frame length, serious inefficiencies result; this is illustrated in Figure 6.2. In the figure, the transmission time (the time it takes for a station to transmit a frame) is normalized to one, and the propagation delay (the time it takes for a bit to travel from sender to receiver) is expressed as the variable a. In other words, when a is less than 1, the propagation time is less than the transmission time. In this case, the frame is sufficiently long that the first bits of the frame have arrived at the destination before the source has completed the transmission of the frame. When a is greater than 1, the propagation time is greater than the transmission time. In this case, the sender completes transmission of the entire frame before the leading bits of that frame arrive at the receiver. Put another way, larger values of a are consistent with higher data rates and/or longer distances between stations. Appendix 6A discusses a and data link performance.

Both parts of the figure (a and b) consist of a sequence of snapshots of the transmission process over time. In both cases, the first four snapshots show the process of transmitting a frame containing data, and the last snapshot shows the return of a small acknowledgment frame. Note that for $a > 1$, the line is always underutilized, and, even for $a < 1$, the line is inefficiently utilized. In essence, for very high data rates, or for very long distances between sender and receiver, stop-and-wait flow control provides inefficient line utilization.

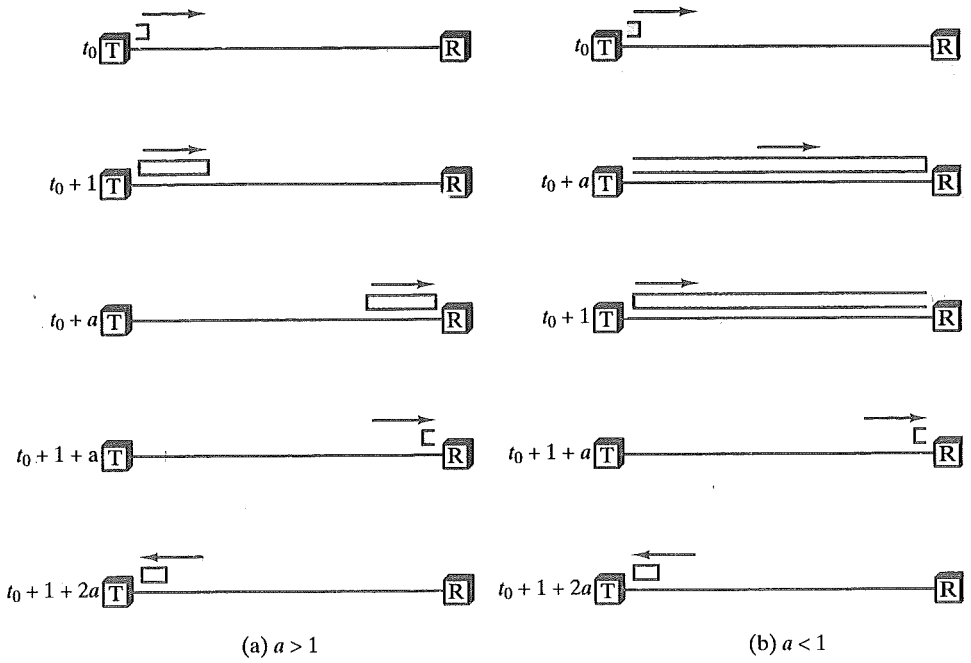


FIGURE 6.2 Stop-and-wait link utilization (transmission time = 1; propagation time = a).

Sliding-Window Flow Control

The essence of the problem described so far is that only one frame at a time can be in transit. In situations where the bit length of the link is greater than the frame length ($a > 1$), serious inefficiencies result. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.

Let us examine how this might work for two stations, A and B, connected via a full-duplex link. Station B allocates buffer space for n frames. Thus, B can accept n frames, and A is allowed to send n frames without waiting for any acknowledgments. To keep track of which frames have been acknowledged, each is labeled with a sequence number. B acknowledges a frame by sending an acknowledgment that includes the sequence number of the next frame expected. This acknowledgment also implicitly announces that B is prepared to receive the next n frames, beginning with the number specified. This scheme can also be used to acknowledge multiple frames. For example, B could receive frames 2, 3, and 4, but withhold acknowledgment until frame 4 has arrived; by then returning an acknowledgment with sequence number 5, B acknowledges frames 2, 3, and 4 at one time. A maintains a list of sequence numbers that it is allowed to send, and B maintains a list of sequence numbers that it is prepared to receive. Each of these lists can be thought of as a *window* of frames. The operation is referred to as sliding-window flow control.

Several additional comments need to be made. Because the sequence number to be used occupies a field in the frame, it is clearly of bounded size. For example, for a 3-bit field, the sequence number can range from 0 to 7. Accordingly, frames are numbered modulo 8; that is, after sequence-number 7, the next number is 0. In general, for a k-bit field the range of sequence numbers is 0 through $2^k - 1$, and frames are numbered modulo 2^k; with this in mind, Figure 6.3 is a useful way of depicting the sliding-window process. It assumes the use of a 3-bit sequence number, so that frames are numbered sequentially from 0 through 7, and then the same numbers are reused for subsequent frames. The shaded rectangle indicates that the sender may transmit 7 frames, beginning with frame 6. Each time a frame is sent, the shaded window shrinks; each time an acknowledgment is received, the shaded window grows.

The actual window size need not be the maximum possible size for a given sequence-number length. For example, using a 3-bit sequence number, a window size of 4 could be configured for the stations using the sliding-window flow control protocol.

An example is shown in Figure 6.4. The example assumes a 3-bit sequence number field and a maximum window size of seven frames. Initially, A and B have windows indicating that A may transmit seven frames, beginning with frame 0 (F0). After transmitting three frames (F0, F1, F2) without acknowledgment, A has shrunk its window to four frames. The window indicates that A may transmit four frames, beginning with frame number 3. B then transmits an RR (receive-ready) 3, which means: "I have received all frames up through frame number 2 and am ready

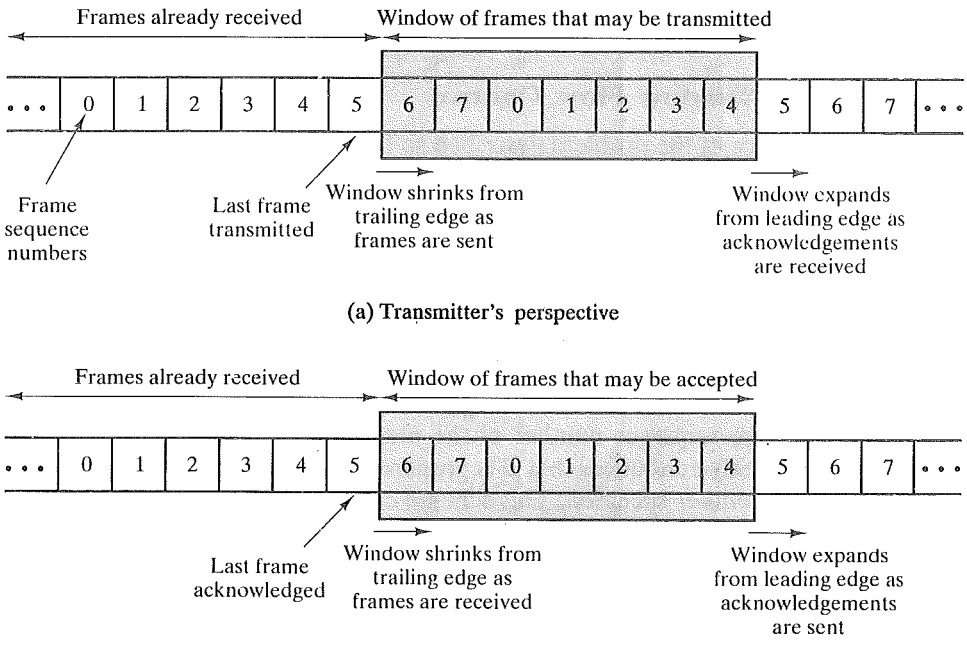


FIGURE 6.3 Sliding-window depiction.

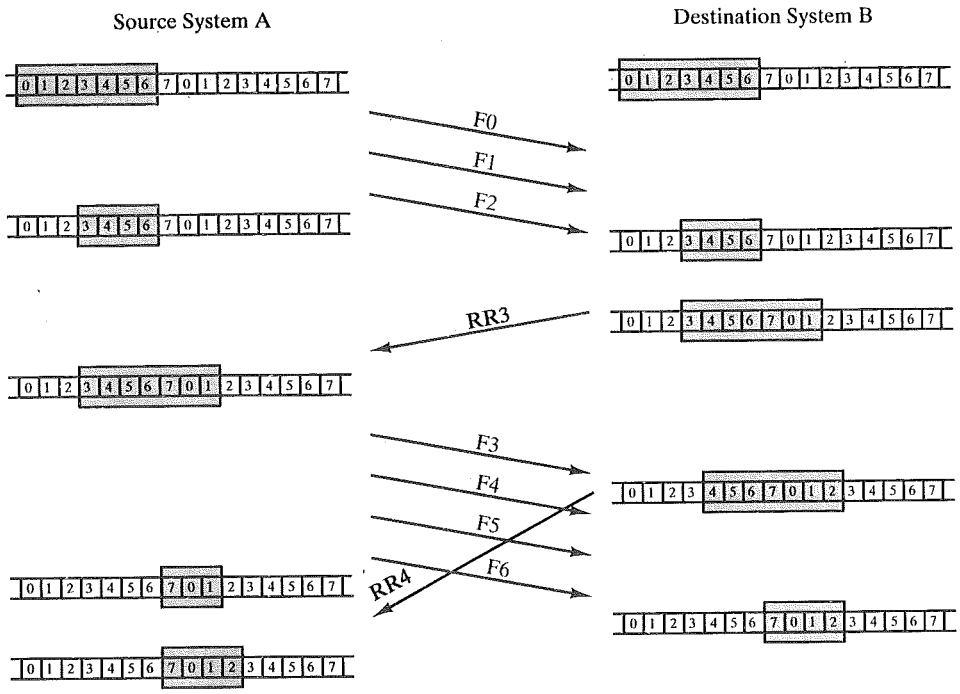


FIGURE 6.4 Example of a sliding-window protocol.

to receive frame number 3; in fact, I am prepared to receive seven frames, beginning with frame number 3." With this acknowledgment, A is back up to permission to transmit seven frames, still beginning with frame 3. A proceeds to transmit frames 3, 4, 5, and 6. B returns an RR 7, which acknowledges all of these frames and permits A to send 7 frames, beginning with frame 7.

The mechanism so far described does indeed provide a form of flow control: The receiver must only be able to accommodate 7 frames beyond the one it has last acknowledged; to supplement this, most protocols also allow a station to completely cut off the flow of frames from the other side by sending a Receive-Not-Ready (RNR) message, which acknowledges former frames but forbids transfer of future frames. Thus, RNR 5 means: "I have received all frames up through number 4 but am unable to accept any more." At some subsequent point, the station must send a normal acknowledgment to reopen the window.

So far, we have discussed transmission in one direction only. If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other. To provide efficient support for this requirement, a feature known as *piggybacking* is typically provided. Each *data frame* includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment. Thus, if a station has data to send and an acknowledgment to send, it sends both together in one frame, thereby saving communication capacity. Of course, if a station has an acknowledgment but no data to send, it sends a separate *acknowledgment frame*. If a station has data to send but no new acknowledgment to send, it

must repeat the last acknowledgment that it sent; this is because the data frame includes a field for the acknowledgment number, and some value must be put into that field. When a station receives a duplicate acknowledgment, it simply ignores it.

It should be clear from the discussion that sliding-window flow control is potentially much more efficient than stop-and-wait flow control. The reason is that, with sliding-window flow control, the transmission link is treated as a pipeline that may be filled with frames in transit. In contrast, with stop-and-wait flow control, only one frame may be in the pipe at a time. Appendix 6A quantifies the improvement in efficiency.

6.2 ERROR DETECTION

In earlier chapters, we talked about transmission impairments and the effect of data rate and signal-to-noise ratio on bit error rate. Regardless of the design of the transmission system, there will be errors, resulting in the change of one or more bits in a transmitted frame.

Let us define these probabilities with respect to errors in transmitted frames:

P_b: Probability of a single bit error; also known as the bit error rate.

P_1: Probability that a frame arrives with no bit errors.

P_2: Probability that a frame arrives with one or more undetected bit errors.

P_3: Probability that a frame arrives with one or more detected bit errors but no undetected bit errors.

First, consider the case when no means are taken to detect errors; the probability of detected errors (P_3), then, is zero. To express the remaining probabilities, assume that the probability that any bit is in error (P_b) is constant and independent for each bit. Then we have

$$P_1 = (1 - P_b)^F$$

$$P_2 = 1 - P_1$$

where F is the number of bits per frame. In words, the probability that a frame arrives with no bit errors decreases when the probability of a single bit error increases, as you would expect. Also, the probability that a frame arrives with no bit errors decreases with increasing frame length; the longer the frame, the more bits it has and the higher the probability that one of these is in error.

Let us take a simple example to illustrate these relationships. A defined object for ISDN connections is that the bit error rate on a 64-kbps channel should be less than 10^{-6} on at least 90% of observed 1-minute intervals. Suppose now that we have the rather modest user requirement that at most one frame with an undetected bit error should occur per day on a continuously used 64-kbps channel, and let us assume a frame length of 1000 bits. The number of frames that can be transmitted in a day comes out to 5.529×10^6, which yields a desired frame error rate of

$P_2 = 1/(5.529 \times 10^6) = 0.18 \times 10^{-6}$. But, if we assume a value of P_b of 10^{-6}, then $P_1 = (0.999999)^{1000} = 0.999$ and, therefore, $P_2 = 10^{-3}$, which is about three orders of magnitude too large to meet our requirement.

This is the kind of result that motivates the use of error-detection techniques. All of these techniques operate on the following principle (Figure 6.5). For a given frame of bits, additional bits that constitute an error-detecting code are added by the transmitter. This code is calculated as a function of the other transmitted bits. The receiver performs the same calculation and compares the two results. A detected error occurs if and only if there is a mismatch. Thus, P_3 is the probability that if a frame contains errors, the error-detection scheme will detect that fact. P_2 is known as the residual error rate, and is the probability that an error will be undetected despite the use of an error-detection scheme.

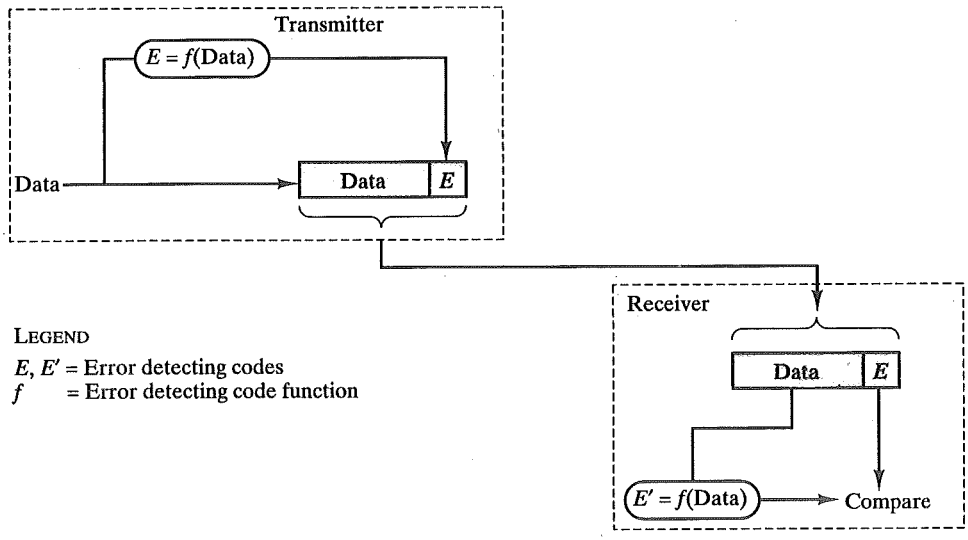


FIGURE 6.5 Error detection.

Parity Check

The simplest error-detection scheme is to append a parity bit to the end of a block of data. A typical example is ASCII transmission, in which a parity bit is attached to each 7-bit ASCII character. The value of this bit is selected so that the character has an even number of 1s (even parity) or an odd number of 1s (odd parity). So, for example, if the transmitter is transmitting an ASCII G (1110001) and using odd parity, it will append a 1 and transmit 11100011. The receiver examines the received character and, if the total number of 1s is odd, assumes that no error has occurred. If one bit (or any odd number of bits) is erroneously inverted during transmission (for example, 11000011), then the receiver will detect an error. Note, however, that if two (or any even number) of bits are inverted due to error, an undetected error occurs. Typically, even parity is used for synchronous transmission and odd parity for asynchronous transmission.

The use of the parity bit is not foolproof, as noise impulses are often long enough to destroy more than one bit, particularly at high data rates.

Cyclic Redundancy Check (CRC)

One of the most common, and one of the most powerful, error-detecting codes is the cyclic redundancy check (CRC), which can be described as follows. Given a k-bit block of bits, or message, the transmitter generates an n-bit sequence, known as a frame check sequence (FCS), so that the resulting frame, consisting of k + n bits, is exactly divisible by some predetermined number. The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.

To clarify this, we present the procedure in three ways: modulo 2 arithmetic, polynomials, and digital logic.

Modulo 2 Arithmetic

Modulo 2 arithmetic uses binary addition with no carries, which is just the exclusive-or operation. For example:

```
  1111          11001
+1010          × 11
------         -----
  0101         11001
              11001
              ------
              101011
```

Now define:

T = (k + n)-bit frame to be transmitted, with n < k
M = k-bit message, the first k bits of T
F = n-bit FCS, the last n bits of T
P = pattern of n + 1 bits; this is the predetermined divisor

We would like T/P to have no remainder. It should be clear that

$$T = 2^n M + F$$

That is, by multiplying M by 2^n, we have, in effect, shifted it to the left by n bits and padded out the result with zeroes. Adding F yields the concatenation of M and F, which is T. We want T to be exactly divisible by P. Suppose that we divided $2^n M$ by P:

$$\frac{2^n M}{P} = Q + \frac{R}{P} \tag{6.1}$$

There is a quotient and a remainder. Because division is binary, the remainder is always at least one bit less than the divisor. We will use this remainder as our FCS. Then

$$T = 2^n M + R$$

Question: Does this R satisfy our condition that T/P have no remainder? To see that it does, consider

$$\frac{T}{P} = \frac{2^n M + R}{P}$$

Substituting Equation (6.1), we have

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

However, any binary number added to itself (modulo 2) yields zero. Thus,

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

There is no remainder, and, therefore, T is exactly divisible by P. Thus, the FCS is easily generated: Simply divide $2^n M$ by P and use the remainder as the FCS. On reception, the receiver will divide T by P and will get no remainder if there have been no errors.

Let us now consider a simple example.

1. Given
   Message M = 1010001101 (10 bits)
   Pattern P = 110101 (6 bits)
   FCS R = to be calculated (5 bits)
2. The message M is multiplied by 2^5, yielding 101000110100000.
3. This product is divided by P:

```
                1101010110 <- Q
P -> 110101 | 101000110100000 <- 2^n M
              110101
               111011
               110101
                 111010
                 110101
                   111110
                   110101
                     101100
                     110101
                      110010
                      110101
                       01110 <- R
```

4. The remainder (R = 01110) is added to $2^n M$ to give T = 101000110101110, which is transmitted.
5. If there are no errors, the receiver receives T intact. The received frame is divided by P:

```
                1101010110←Q
P→110101|1010001101011110←T
        110101
         111011
         110101
           111010
           110101
             111110
             110101
               101111
               110101
                110101
                110101
                 00000←R
```

Because there is no remainder, it is assumed that there have been no errors.

The pattern P is chosen to be one bit longer than the desired FCS, and the exact bit pattern chosen depends on the type of errors expected. At minimum, both the high- and low-order bits of P must be 1.

The occurrence of an error is easily expressed. An error results in the reversal of a bit. This is equivalent to taking the exclusive-or of the bit and 1 (modulo 2 addition of 1 to the bit): $0 + 1 = 1$; $1 + 1 = 0$. Thus, the errors in an $(n + k)$-bit frame can be represented by an $(n + k)$-bit field with 1s in each error position. The resulting frame T_r can be expressed as

$$T_r = T + E$$

where

T = transmitted frame
E = error pattern with 1s in positions where errors occur
T_r = received frame

The receiver will fail to detect an error if and only if T_r is divisible by P, which is equivalent to E divisible by P. Intuitively, this seems an unlikely occurrence.

Polynomials

A second way of viewing the CRC process is to express all values as polynomials in a dummy variable X, with binary coefficients. The coefficients correspond to the bits in the binary number. Thus, for $M = 110011$, we have $M(X) = X^5 + X^4 + X + 1$, and, for $P = 11001$, we have $P(X) = X^4 + X^3 + 1$. Arithmetic operations are again modulo 2. The CRC process can now be described as

$$\frac{X^n M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$T(X) = X^n M(X) + R(X)$$

An error $E(X)$ will only be undetectable if it is divisible by $P(X)$. It can be shown [PETE61] that all of the following errors are not divisible by a suitably chosen $P(X)$ and, hence, are detectable:

- All single-bit errors.
- All double-bit errors, as long as $P(X)$ has at least three 1s.
- Any odd number of errors, as long as $P(X)$ contains a factor $(X + 1)$.
- Any burst error for which the length of the burst is less than the length of the divisor polynomial; that is, less than or equal to the length of the FCS.
- Most larger burst errors.

In addition, it can be shown that if all error patterns are considered equally likely, then for a burst error of length $r + 1$, the probability that $E(X)$ is divisible by $P(X)$ is $1/2^{r-1}$, and for a longer burst, the probability is $1/2^r$, where r is the length of the FCS.

Three versions of $P(X)$ are widely used:

$$\begin{aligned}
\text{CRC-16} &= X^{16} + X^{15} + X^2 + 1 \\
\text{CRC-CCITT} &= X^{16} + X^{12} + X^5 + 1 \\
\text{CRC-32} &= X^{32} + X^{26} + X + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} \\
&\quad + X^8 + X^7 + X^5 + X^4 + X^2 + 1
\end{aligned}$$

Digital Logic

The CRC process can be represented by, and indeed implemented as, a dividing circuit consisting of exclusive-or gates and a shift register. The shift register is a string of 1-bit storage devices. Each device has an output line, that indicates the value currently stored, and an input line. At discrete time instants, known as clock times, the value in the storage device is replaced by the value indicated by its input line. The entire register is clocked simultaneously, causing a 1-bit shift along the entire register.

The circuit is implemented as follows:

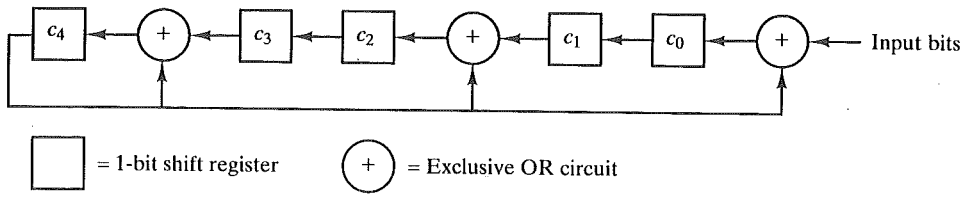
1. The register contains n bits, equal to the length of the FCS.
2. There are up to n exclusive-or gates.
3. The presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial, $P(X)$.

The architecture of this circuit is best explained by first considering an example, which is illustrated in Figure 6.6. In this example, we use

$$\begin{aligned}
\text{Message } M &= 1010001101; & M(X) &= X^9 + X^7 + X^3 + X^2 + 1 \\
\text{Divisor } P &= 110101; & P(X) &= X^5 + X^4 + X^2 + 1
\end{aligned}$$

which were used earlier in the discussion.

Part (a) of the figure shows the shift register implementation. The process begins with the shift register cleared (all zeros). The message, or dividend, is then



(a) Shift-register implementation

	c_4	c_3	c_2	c_1	c_0	$c_4 \oplus c_3$	$c_4 \oplus c_1$	$c_4 \oplus$ input	input	
Initial	0	0	0	0	0	0	0	1	1	
Step 1	0	0	0	0	1	0	0	0	0	
Step 2	0	0	0	1	0	0	1	1	1	
Step 3	0	0	1	0	1	0	0	0	0	
Step 4	0	1	0	1	0	1	1	0	0	Message to be sent
Step 5	1	0	1	0	0	1	1	1	0	
Step 6	1	1	1	0	1	0	1	0	1	
Step 7	0	1	1	1	0	1	1	1	1	
Step 8	1	1	1	0	1	0	1	1	0	
Step 9	0	1	1	1	1	1	1	1	1	
Step 10	1	1	1	1	1	0	0	1	0	
Step 11	0	1	0	1	1	1	1	0	0	
Step 12	1	0	1	1	0	1	0	1	0	Five zeros added
Step 13	1	1	0	0	1	0	1	1	0	
Step 14	0	0	1	1	1	0	1	0	0	
Step 15	0	1	1	1	0	1	1	0	—	

FIGURE 6.6 Circuit with shift registers for dividing by the polynomial $X^5 + X^4 + X^2 + 1$.

entered, one bit at a time, starting with the most significant bit. Part (b) is a table that shows the step-by-step operation as the input is applied one bit at a time. Each row of the table shows the values currently stored in the five shift-register elements. In addition, the row shows the values that appear at the outputs of the three exclusive-or circuits. Finally, the row shows the value of the next input bit, which is available for the operation of the next step.

Because no feedback occurs until a 1-dividend bit arrives at the most significant end of the register, the first five operations are simple shifts. Whenever a 1 bit arrives at the left end of the register (c_4), a 1 is subtracted (exclusive-or) from the second (c_3), fourth (c_1), and sixth (input) bits on the next shift. This is identical to the binary long-division process illustrated earlier. The process continues through all the bits of the message, plus five zero bits. These latter bits account for shifting M to the left five position to accommodate the FCS. After the last bit is processed, the shift register contains the remainder (FCS), which can then be transmitted.

At the receiver, the same logic is used. As each bit of M arrives, it is inserted into the shift register. If there have been no errors, the shift register should contain

the bit pattern for R at the conclusion of M. The transmitted bits of R now begin to arrive, and the effect is to zero-out the register so that, at the conclusion of reception, the register contains all 0s.

Figure 6.7 indicates the general architecture of the shift register implementation of a CRC for the polynomial $P(X) = \sum_{i=0}^{n} a_i X^i$ where $a_0 = a_n = 1$ and all other a_i equal either 0 or 1.

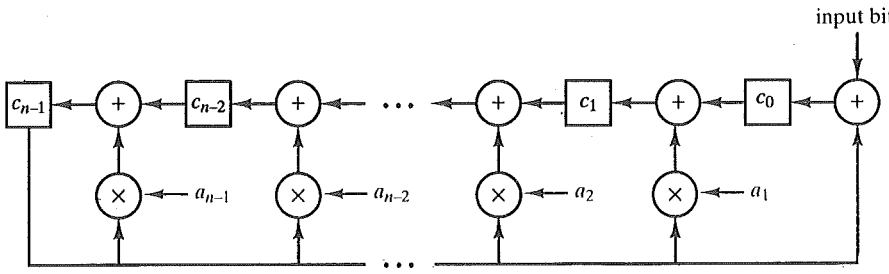


FIGURE 6.7 General CRC architecture to implement divisor $1 + a_1X + a_2X^2 + \cdots + a_{n-1}X^{n-1} + X^n$.

6.3 ERROR CONTROL

Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames. The model that we will use, which covers the typical case, is illustrated in Figure 6.1b. As before, data are sent as a sequence of frames; frames arrive in the same order in which they are sent; and each transmitted frame suffers an arbitrary and variable amount of delay before reception. In addition, we admit the possibility of two types of errors:

- **Lost frame.** A frame fails to arrive at the other side. For example, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
- **Damaged frame.** A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).

The most common techniques for error control are based on some or all of the following ingredients:

- **Error detection.** As discussed in the preceding section.
- **Positive acknowledgment.** The destination returns a positive acknowledgment to successfully received, error-free frames.

- **Retransmission after timeout.** The source retransmits a frame that has not been acknowledged after a predetermined amount of time.
- **Negative acknowledgment and retransmission.** The destination returns a negative acknowledgment to frames in which an error is detected. The source retransmits such frames.

Collectively, these mechanisms are all referred to as **automatic repeat request** (ARQ); the effect of ARQ is to turn an unreliable data link into a reliable one. Three versions of ARQ have been standardized:

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ

All of these forms are based on the use of the flow control technique discussed in Section 6.1. We examine each in turn.

Stop-and-Wait ARQ

Stop-and-wait ARQ is based on the stop-and-wait flow-control technique outlined previously and is depicted in Figure 6.8. The source station transmits a single frame and then must await an acknowledgment (ACK). No other data frames can be sent until the destination station's reply arrives at the source station.

Two sorts of errors could occur. First, the frame that arrives at the destination could be damaged; the receiver detects this by using the error detection technique referred to earlier and simply discards the frame. To account for this possibility, the source station is equipped with a timer. After a frame is transmitted, the source station waits for an acknowledgment. If no acknowledgment is received by the time the timer expires, then the same frame is sent again. Note that this method requires that the transmitter maintain a copy of a transmitted frame until an acknowledgment is received for that frame.

The second sort of error is a damaged acknowledgment. Consider the following situation. Station *A* sends a frame. The frame is received correctly by station *B*, which responds with an acknowledgment (ACK). The ACK is damaged in transit and is not recognizable by *A*, which will therefore time-out and resend the same frame. This duplicate frame arrives and is accepted by *B*, which has therefore accepted two copies of the same frame as if they were separate. To avoid this problem, frames are alternately labeled with 0 or 1, and positive acknowledgments are of the form ACK0 and ACK1. In keeping with the sliding-window convention, an ACK0 acknowledges receipt of a frame numbered 1 and indicates that the receiver is ready for a frame numbered 0.

The principal advantage of stop-and-wait ARQ is its simplicity. Its principal disadvantage, as discussed in Section 6.1, is that stop-and-wait is an inefficient mechanism. The sliding-window flow control technique can be adapted to provide more efficient line use; in this context, it is sometimes referred to as *continuous ARQ*.

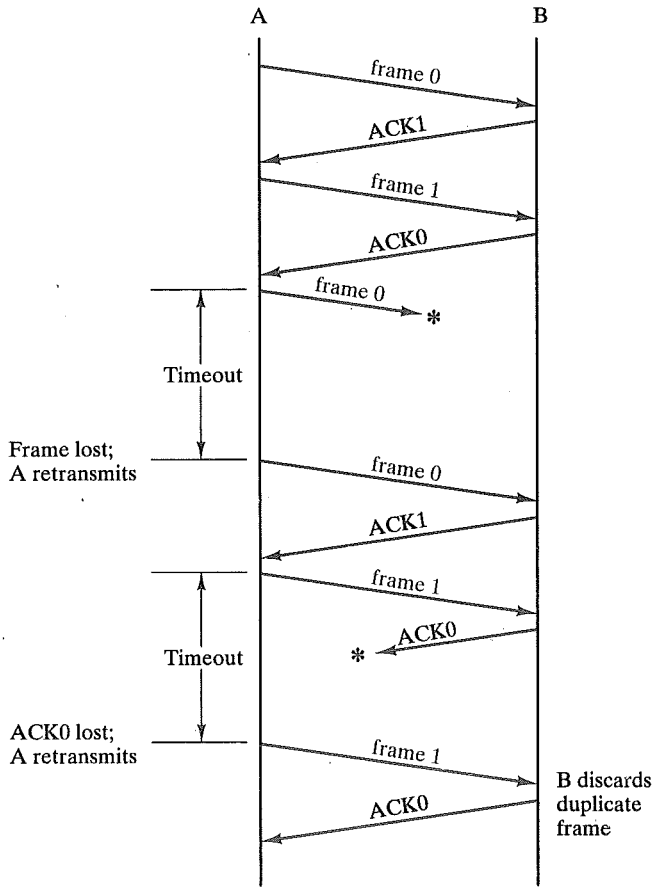


FIGURE 6.8 Stop-and-wait ARQ.

Go-back-N ARQ

The form of error control based on sliding-window flow control that is most commonly used is called go-back-N ARQ.

In go-back-N ARQ, a station may send a series of frames sequentially numbered modulo some maximum value. The number of unacknowledged frames outstanding is determined by window size, using the sliding-window flow control technique. While no errors occur, the destination will acknowledge (RR = receive-ready) incoming frames as usual. If the destination station detects an error in a frame, it sends a negative acknowledgment (REJ = reject) for that frame. The destination station will discard that frame and all future incoming frames until the frame in error is correctly received. Thus, the source station, when it receives an REJ, must retransmit the frame in error plus all succeeding frames that were transmitted in the interim.

Consider that station A is sending frames to station B. After each transmission, A sets an acknowledgment timer for the frame just transmitted. The go-back-N technique takes into account the following contingencies:

1. Damaged frame. There are three subcases:
   a) A transmits frame i. B detects an error and has previously successfully received frame $(i - 1)$. B sends REJ i, indicated that frame i is rejected. When A receives the REJ, it must retransmit frame i and all subsequent frames that it has transmitted since the original transmission of frame i.
   b) Frame i is lost in transit. A subsequently sends frame $(i + 1)$. B receives frame $(i + 1)$ out of order and sends an REJ i. A must retransmit frame i and all subsequent frames.
   c) Frame i is lost in transit, and A does not soon send additional frames. B receives nothing and returns neither an RR nor an REJ. When A's timer expires, it transmits an RR frame that includes a bit known as the P bit, which is set to 1. B interprets the RR frame with a P bit of 1 as a command that must be acknowledged by sending an RR indicating the next frame that it expects. When A receives the RR, it retransmits frame i.
2. Damaged RR. There are two subcases:
   a) B receives frame i and sends RR $(i + 1)$, which is lost in transit. Because acknowledgments are cumulative (e.g., RR 6 means that all frames through 5 are acknowledged), it may be that A will receive a subsequent RR to a subsequent frame and that it will arrive before the timer associated with frame i expires.
   b) If A's timer expires, it transmits an RR command as in Case 1b. It sets another timer, called the P-bit timer. If B fails to respond to the RR command, or if its response is damaged, then A's P-bit timer will expire. At this point, A will try again by issuing a new RR command and restarting the P-bit timer. This procedure is tried for a number of iterations. If A fails to obtain an acknowledgment after some maximum number of attempts, it initiates a reset procedure.
3. Damaged REJ. If an REJ is lost, this is equivalent to Case 1c.

Figure 6.9 is an example of the frame flow for go-back-N ARQ. Because of the propagation delay on the line, by the time that an acknowledgment (positive or negative) arrives back at the sending station, it has already sent two additional frames beyond the one being acknowledged. Thus, when an REJ is received to frame 5, not only frame 5, but frames 6 and 7, must be retransmitted. Thus, the transmitter must keep a copy of all unacknowledged frames.

In Section 6.1, we mentioned that for a k-bit sequence number field, which provides a sequence number range of 2^k, the maximum window size is limited to $2^k - 1$. This has to do with the interaction between error control and acknowledgment. Consider that if data are being exchanged in both directions, station B must send piggybacked acknowledgments to station A's frames in the data frames being transmitted by B, even if the acknowledgment has already been sent; as we have mentioned, this is because B must put some number in the acknowledgment field of

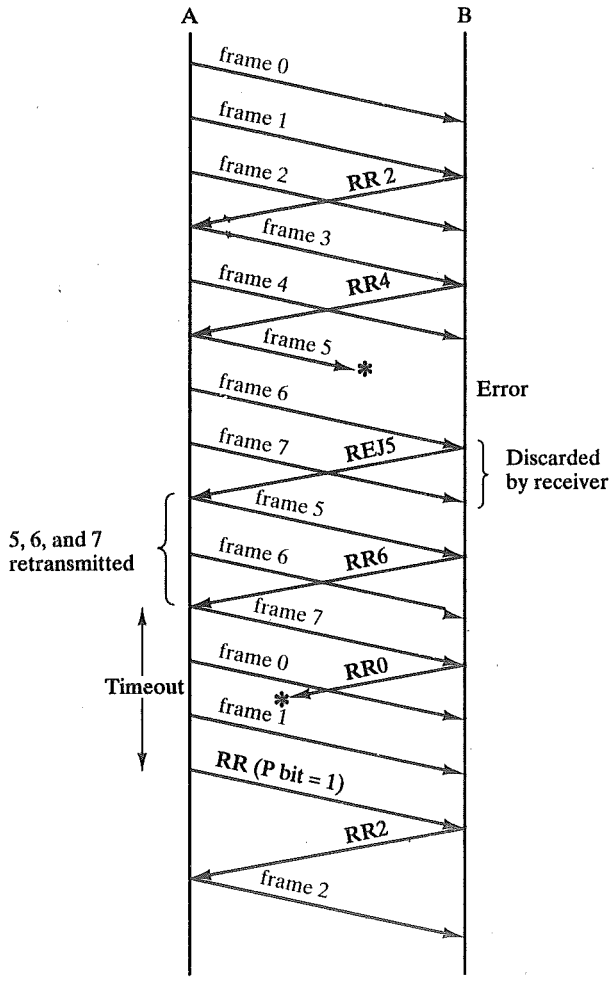


FIGURE 6.9 Go-back-N ARQ.

its data frame. As an example, assume a 3-bit sequence number (sequence-number space = 8). Suppose a station sends frame 0 and gets back an RR 1, and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR 1. This could mean that all eight frames were received correctly and the RR 1 is a cumulative acknowledgment. It could also mean that all eight frames were damaged or lost in transit, and the receiving station is repeating its previous RR 1. The problem is avoided if the maximum window size is limited to 7 ($2^3 - 1$).

Selective-reject ARQ

With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment, in this case called SREJ, or that time-out. This would

appear to be more efficient than go-back-N, because it minimizes the amount of retransmission. On the other hand, the receiver must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted, and it must contain logic for reinserting that frame in the proper sequence. The transmitter, too, requires more complex logic to be able to send a frame out of sequence. Because of such complications, select-reject ARQ is much less used than go-back-N ARQ.

The window-size limitation is more restrictive for selective-reject than for go-back-N. Consider the case of a 3-bit sequence-number size for selective-reject. Allow a window size of seven, and consider the following scenario [TANE88]:

1. Station *A* sends frames 0 through 6 to station *B*.
2. Station *B* receives all seven frames and cumulatively acknowledges with RR 7.
3. Because of a noise burst, the RR 7 is lost.
4. *A* times out and retransmits frame 0.
5. *B* has already advanced its receive window to accept frames 7, 0, 1, 2, 3, 4, and 5. Thus, it assumes that frame 7 has been lost and that this is a new frame 0, which it accepts.

The problem with the foregoing scenario is that there is an overlap between the sending and receiving windows. To overcome the problem, the maximum window size should be no more than half the range of sequence numbers. In the scenario above, if only four unacknowledged frames may be outstanding, no confusion can result. In general, for a k-bit sequence number field, which provides a sequence number range of 2^k, the maximum window size is limited to 2^{k-1}.

6.4 HIGH-LEVEL DATA LINK CONTROL (HDLC)

The most important data link control protocol is HDLC (ISO 33009, ISO 4335). Not only is HDLC widely used, but it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC. Accordingly, in this section we provide a detailed discussion of HDLC. Section 6.5 surveys related protocols.

Basic Characteristics

To satisfy a variety of applications, HDLC defines three types of stations, two link configurations, and three data-transfer modes of operation. The three station types are

- **Primary station.** Has the responsibility for controlling the operation of the link. Frames issued by the primary are called *commands*.
- **Secondary station.** Operates under the control of the primary station. Frames

issued by a secondary are called *responses*. The primary maintains a separate logical link with each secondary station on the line.

- **Combined station.** Combines the features of primary and secondary. A combined station may issue both commands and responses.

The two link configurations are

- **Unbalanced configuration.** Consists of one primary and one or more secondary stations and supports both full-duplex and half-duplex transmission.
- **Balanced configuration.** Consists of two combined stations and supports both full-duplex and half-duplex transmission.

The three data transfer modes are

- **Normal response mode (NRM).** Used with an unbalanced configuration. The primary may initiate data transfer to a secondary, but a secondary may only transmit data in response to a command from the primary.
- **Asynchronous balanced mode (NRM).** Used with a balanced configuration. Either combined station may initiate transmission without receiving permission from the other combined station.
- **Asynchronous response mode (NRM).** Used with an unbalanced configuration. The secondary may initiate transmission without explicit permission of the primary. The primary still retains responsibility for the line, including initialization, error recovery, and logical disconnection.

NRM is used on mulitdrop lines, in which a number of terminals are connected to a host computer. The computer polls each terminal for input. NRM is also sometimes used on point-to-point links, particularly if the link connects a terminal or other peripheral to a computer. ABM is the most widely used of the three modes; it makes more efficient use of a full-duplex point-to-point link as there is no polling overhead. ARM is rarely used; it is applicable to some special situations in which a secondary may need to initiate transmission.

Frame Structure

HDLC uses synchronous transmission. All transmissions are in the form of frames, and a single frame format suffices for all types of data and control exchanges.

Figure 6.10a depicts the structure of the HDLC frame. The flag, address, and control fields that precede the information field are known as a header. The FCS and flag fields following the data field are referred to as a *trailer*.

Flag Fields

Flag fields delimit the frame at both ends with the unique pattern 01111110. A single flag may be used as the closing flag for one frame and the opening flag for the next. On both sides of the user-network interface, receivers are continuously hunting for the flag sequence to synchronize on the start of a frame. While receiving a

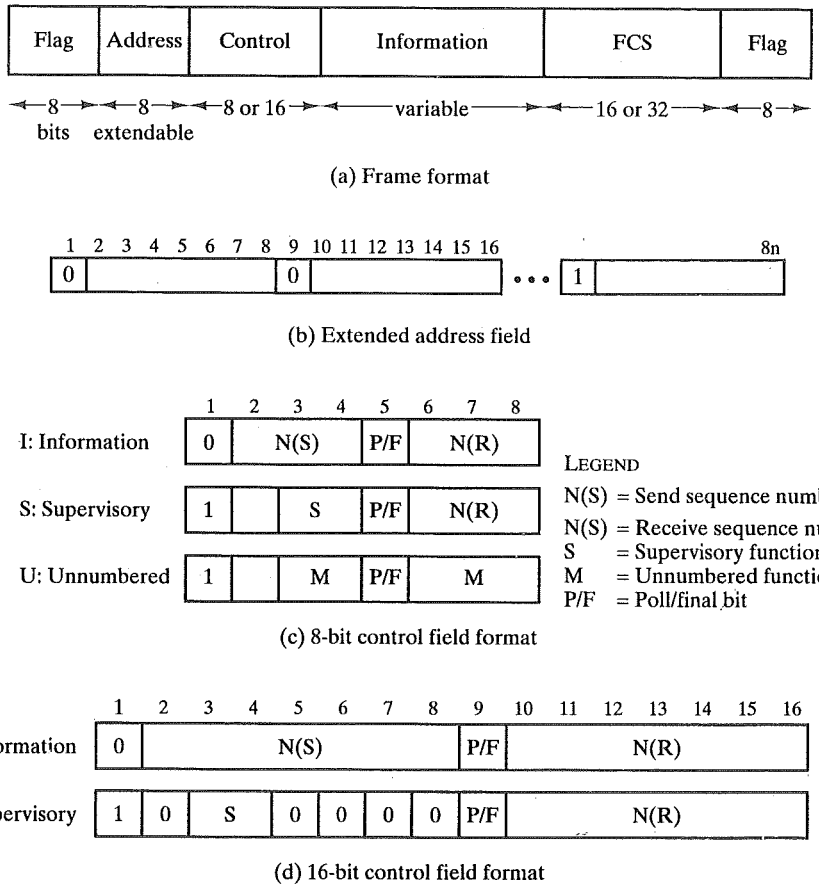


FIGURE 6.10 HDLC frame structure.

frame, a station continues to hunt for that sequence to determine the end of the frame. However, it is possible that the pattern 01111110 will appear somewhere inside the frame, thus destroying frame-level synchronization. To avoid this, a procedure known as *bit stuffing* is used. Between the transmission of the starting and ending flags, the transmitter will always insert an extra 0 bit after each occurrence of five 1s in the frame. After detecting a starting flag, the receiver monitors the bit stream. When a pattern of five 1s appears, the sixth bit is examined. If this bit is 0, it is deleted. If the sixth bit is a 1 and the seventh bit is a 0, the combination is accepted as a flag. If the sixth and seventh bits are both 1, the sender is indicating an abort condition.

With the use of bit stuffing, arbitrary bit patterns can be inserted into the data field of the frame. This property is known as *data transparency*.

Figure 6.11 shows an example of bit stuffing. Note that in the first two cases, the extra 0 is not strictly necessary for avoiding a flag pattern, but is necessary for the operation of the algorithm. The pitfalls of bit stuffing are also illustrated in this

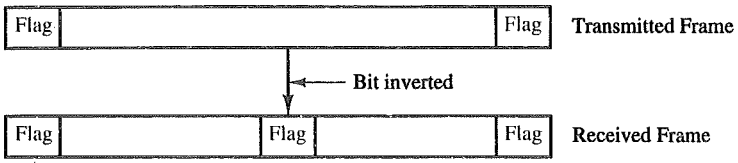
Original pattern

111111111111011111101111110

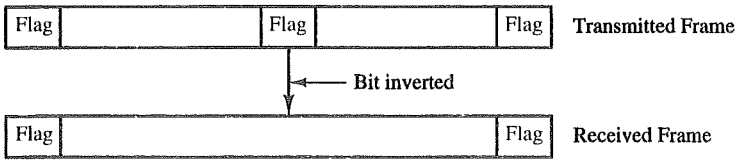
After bit-stuffing

11111▮11111▮1101111101▮11111▮10

(a) Example



(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

FIGURE 6.11 Bit stuffing.

figure. When a flag is used as both an ending and a starting flag, a 1-bit error merges two frames into one; conversely, a 1-bit error inside the frame could split it in two.

Address Field

The address field identifies the secondary station that transmitted or is to receive the frame. This field is not needed for point-to-point links, but is always included for the sake of uniformity. The address field is usually eight bits long but, by prior agreement, an extended format may be used in which the actual address length is a multiple of seven bits (Figure 6.10b). The least significant bit of each octet is 1 or 0, depending on whether it is or is not the last octet of the address field. The remaining seven bits of each octet form part of the address. The single-octet address of 11111111 is interpreted as the all-stations address in both basic and extended formats. It is used to allow the primary to broadcast a frame for reception by all secondaries.

Control Field

HDLC defines three types of frames, each with a different control field format. *Information frames* (I-frames) carry the data to be transmitted for the user (the logic above HDLC that is using HDLC). Additionally, flow- and error-control data,

using the ARQ mechanism, are piggybacked on an information frame. *Supervisory frames* (S-frames) provide the ARQ mechanism when piggybacking is not used. *Unnumbered frames* (U-frames) provide supplemental link control functions. The first one or two bits of the control field serves to identify the frame type. The remaining bit positions are organized into subfields as indicated in Figure 6.10c and d. Their use is explained below in the discussion of HDLC operation.

Note that the basic control field for S- and I-frames uses 3-bit sequence numbers. With the appropriate set-mode command, an extended control field can be used for S- and I-frames that employs 7-bit sequence numbers. U-frames always contain an 8-bit control field.

Information Field

The information field is present only in I-frames and some U-frames. The field can contain any sequence of bits but must consist of an integral number of octets. The length of the information field is variable up to some system-defined maximum.

Frame Check Sequence Field

The frame check sequence (FCS) is an error-detecting code calculated from the remaining bits of the frame, exclusive of flags. The normal code is the 16-bit CRC-CCITT defined in Section 6.2. An optional 32-bit FCS, using CRC-32, may be employed if the frame length or the line reliability dictates this choice.

Operation

HDLC operation consists of the exchange of I-frames, S-frames, and U-frames between two stations. The various commands and responses defined for these frame types are listed in Table 6.1. In describing HDLC operation, we will discuss these three types of frames.

The operation of HDLC involves three phases. First, one side or another initializes the data link so that frames may be exchanged in an orderly fashion. During this phase, the options that are to be used are agreed upon. After initialization, the two sides exchange user data and the control information to exercise flow and error control. Finally, one of the two sides signals the termination of the operation.

Initialization

Initialization may be requested by either side by issuing one of the six set-mode commands. This command serves three purposes:

1. It signals the other side that initialization is requested.
2. It specifies which of the three modes (NRM, ABM, ARM) is requested.
3. It specifies whether 3- or 7-bit sequence numbers are to be used.

If the other side accepts this request, then the HDLC module on that end transmits an unnumbered acknowledged (UA) frame back to the initiating side. If the request is rejected, then a disconnected mode (DM) frame is sent.

TABLE 6.1 HDLC Commands and responses.

Name	Command/ response	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
Unnumbered (U)		
Set normal response/extended mode (SNRM/SNRME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM, SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	C	Terminate logical link connection
Request disconnect (RD)	R	Request for DISC command
Request initialization mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered information (UI)	C/R	Used to exchange control information
Unnumbered poll (UP)	C	Used to solicit control information
Reset (RSET)	C	Used for recovery; resets N(R), N(S)
Exchange identification (XID)	C/R	Used to request/report status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Reports receipt of unacceptable frame

Data Transfer

When the initialization has been requested and accepted, then a logical connection is established. Both sides may begin to send user data in I-frames, starting with sequence number 0. The N(S) and N(R) fields of the I-frame are sequence numbers that support flow control and error control. An HDLC module sending a sequence of I-frames will number them sequentially, modulo 8 or 128, depending on whether 3- or 7-bit sequence numbers are used, and place the sequence number in N(S). N(R) is the acknowledgment for I-frames received; it enables the HDLC module to

indicate which number I-frame it expects to receive next.

S-frames are also used for flow control and error control. The receive-ready (RR) frame is used to acknowledge the last I-frame received by indicating the next I-frame expected. The RR is used when there is no reverse-user data traffic (I-frames) to carry an acknowledgment. Receive-not-ready (RNR) acknowledges an I-frame, as with RR, but also asks the peer entity to suspend transmission of I-frames. When the entity that issued RNR is again ready, it sends an RR. REJ initiates the go-back-N ARQ. It indicates that the last I-frame received has been rejected and that retransmission of all I-frames beginning with number N(R) is required. Selective reject (SREJ) is used to request retransmission of just a single frame.

Disconnect

Either HDLC module can initiate a disconnect, either on its own initiative if there is some sort of fault, or at the request of its higher-layer user. HDLC issues a disconnect by sending a disconnect (DISC) frame. The other side must accept the disconnect by replying with a UA.

Examples of Operation

In order to better understand HDLC operation, several examples are presented in Figure 6.12. In the example diagrams, each arrow includes a legend that specifies the frame name, the setting of the P/F bit, and, where appropriate, the values of N(R) and N(S). The setting of the P or F bit is 1 if the designation is present and 0 if absent.

Figure 6.12a shows the frames involved in link setup and disconnect. The HDLC entity for one side issues an SABM command to the other side and starts a timer. The other side, upon receiving the SABM, returns a UA response and sets local variables and counters to their initial values. The initiating entity receives the UA response, sets its variables and counters, and stops the timer. The logical connection is now active, and both sides may begin transmitting frames. Should the timer expire without a response, the originator will repeat the SABM, as illustrated. This would be repeated until a UA or DM is received or until, after a given number of tries, the entity attempting initiation gives up and reports failure to a management entity. In such a case, higher-layer intervention is necessary. The same figure (Figure 6.12a) shows the disconnect procedure. One side issues a DISC command, and the other responds with a UA response.

Figure 6.12b illustrates the full-duplex exchange of I-frames. When an entity sends a number of I-frames in a row with no incoming data, then the receive sequence number is simply repeated (e.g., I, 1, 1; I, 2, 1 in the A-to-B direction). When an entity receives a number of I-frames in a row with no outgoing frames, then the receive sequence number in the next outgoing frame must reflect the cumulative activity (e.g., I, 1, 3 in the B-to-A direction). Note that, in addition to I-frames, data exchange may involve supervisory frames.

Figure 6.12c shows an operation involving a busy condition. Such a condition may arise because an HDLC entity is not able to process I-frames as fast as they are

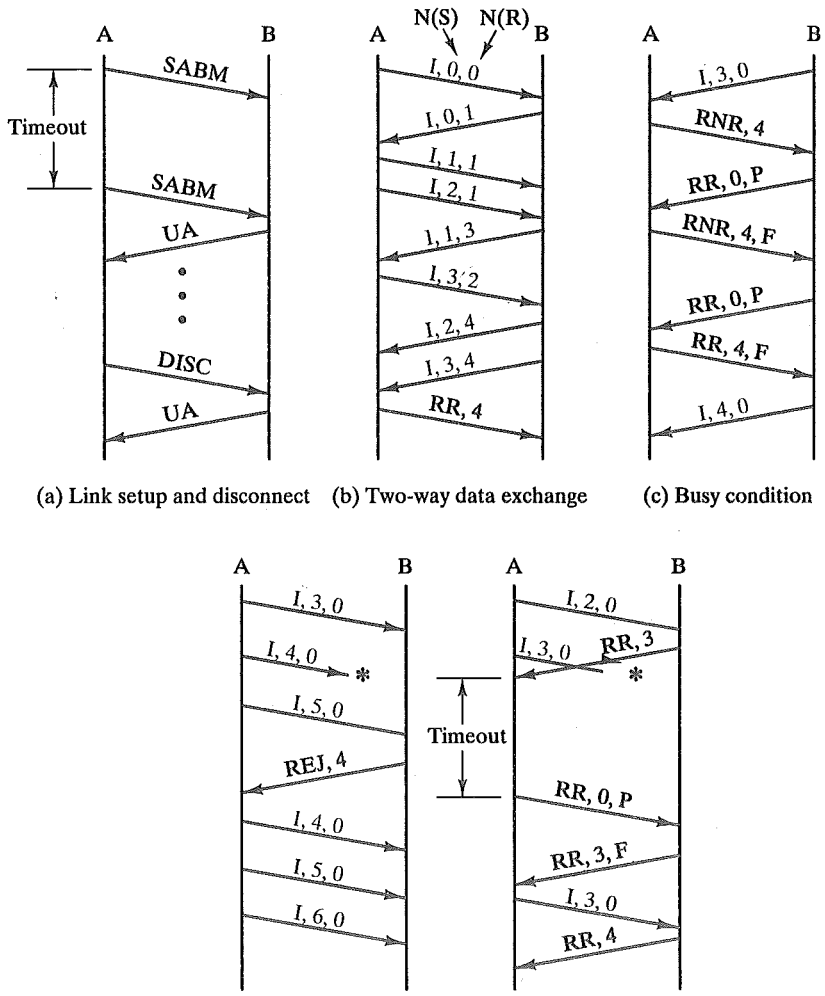


FIGURE 6.12 Examples of HDLC operation.

arriving, or the intended user is not able to accept data as fast as they arrive in I-frames. In either case, the entity's receive buffer fills up and it must halt the incoming flow of I-frames, using an RNR command. In this example, A issues an RNR, which requires B to halt transmission of I-frames. The station receiving the RNR will usually poll the busy station at some periodic interval by sending an RR with the P-bit set; this requires the other side to respond with either an RR or an RNR. When the busy condition has cleared, A returns an RR, and I-frame transmission from B can resume.

An example of error recovery using the REJ command is shown in Figure 6.12d. In this example, A transmits I-frames numbered 3, 4, and 5. Number 4 suffers an error and is lost. When B receives I-frame number 5, it discards this frame

Flag	Address	Control	Information	FCS	Flag
8	8n	8 or 16	Variable	16 or 32	8

(a) HDLC, LAPB

Flag	Address	Control	Information	FCS	Flag
8	16	16*	Variable	16	8

(b) LAPD

MAC control	Dest. MAC address	Source MAC address	DSAP	SSAP	LLC control	Information	FCS
Variable	16 or 48	16 or 48	8	8	16*	Variable	32

(c) LLC/MAC

Flag	Address	Control	Information	FCS	Flag
8	16 or 32	16*	Variable	16 or 32	8

(d) LAPF (control)

Flag	Address	Information	FCS	Flag
8	16 to 32	Variable	16	8

(e) LAPF (core)

General flow control	Virtual path identifier	Virtual channel identifier	Control bits	Header error control	Information
4	8	16	4	8	384

(f) ATM

* = 16-bit control field (7 bit sequence numbers) for I- and S-frames; 8 bit for U-frames.

FIGURE 6.13 Data link control frame formats.

Logical Link Control (LLC)

LLC is part of the IEEE 802 family of standards for controlling operation over a local area network (LAN). LLC is lacking some features found in HDLC and also has some features not found in HDLC.

The most obvious difference between LLC and HDLC is the difference in frame format. Link control functions in the case of LLC are actually divided between two layers: a medium access control (MAC) layer, and the LLC layer, which operates on top of the MAC layer.

Figure 6.13c shows the structure of the combined MAC/LLC frame; the shaded portion corresponds to the fields produced at the LLC layer, and the unshaded portions are the header and trailer of the MAC frame. The MAC layer includes source and destination addresses for devices attached to the LAN. Two addresses are needed as there is no concept of primary and secondary in the LAN environment; therefore, both the sender and receiver must be identified. Error

because it is out of order and sends an REJ with an N(R) of 4. This causes A to initiate retransmission of all I-frames sent, beginning with frame 4. It may continue to send additional frames after the retransmitted frames.

An example of error recovery using a timeout is shown in Figure 6.12e. In this example, A transmits I-frame number 3 as the last in a sequence of I-frames. The frame suffers an error. B detects the error and discards it. However, B cannot send an REJ; this is because there is no way to know if this was an I-frame. If an error is detected in a frame, all of the bits of that frame are suspect, and the receiver has no way to act upon it. A, however, would have started a timer as the frame was transmitted. This timer has a duration long enough to span the expected response time. When the timer expires, A initiates recovery action; this is usually done by polling the other side with an RR command with the P bit set, to determine the status of the other side. Because the poll demands a response, the entity will receive a frame containing an N(R) field and be able to proceed. In this case, the response indicates that frame 3 was lost, which A retransmits.

These examples are not exhaustive. However, they should give the reader a good feel for the behavior of HDLC.

6.5 OTHER DATA LINK CONTROL PROTOCOLS

In addition to HDLC, there are a number of other important data link control protocols. Figure 6.13 illustrates the frame formats, and this section provides a brief overview.

LAPB

LAPB (Link Access Procedure, Balanced) was issued by ITU-T as part of its X.25 packet-switching network-interface standard. It is a subset of HDLC that provides only the asynchronous balanced mode (ABM); it is designed for the point-to-point link between a user system and a packet-switching network node. Its frame format is the same as that of HDLC.

LAPD

LAPD (Link Access Procedure, D-Channel) was issued by ITU-T as part of its set of recommendations on ISDN (Integrated Services Digital Network). LAPD provides data link control over the D channel, which is a logical channel at the user-ISDN interface.

There are several key differences between LAPD and HDLC. Like LAPB, LAPD is restricted to ABM. LAPD always uses 7-bit sequence numbers; 3-bit sequence numbers are not allowed. The FCS for LAPD is always the 16-bit CRC. Finally, the address field for LAPD is a 16-bit field that actually contains two subaddresses: one is used to identify one of possibly multiple devices on the user side of the interface, and the other is used to identify one of possibly multiple logical users of LAPD on the user side of the interface.

detection is done at the MAC level, using a 32-bit CRC. Finally, there are some control functions peculiar to medium-access control that may be included in a MAC control field.

At the LLC layer, there are four fields. The destination and source service access points (DSAP and SSAP), identify the logical user of LLC at the source and destination systems. The LLC control field has the same format as that of HDLC, limited to 7-bit sequence numbers.

Operationally, LLC offers three forms of *service*. The connection-mode service is the same as the ABM of HDLC. The other two services, unacknowledged connectionless and acknowledged connectionless, are described in Part II.

Frame Relay

Frame relay is a data link control facility designed to provide a streamlined capability for use over high-speed packet-switched networks. It is used in place of X.25, which consists of both a data link control protocol (LAPB) and a network-layer protocol (called X.25 packet layer). Frame relay is examined in detail in Part II.

The data link control protocol defined for frame relay is LAPF (Link Access Procedure for Frame-Mode Bearer Services). There are actually two protocols: a *control protocol*, which has similar features to HDLC, and a *core protocol*, which is a subset of the control protocol.

There are several key differences between the LAPF control protocol and HDLC. Like LAPB, LAPF control is restricted to ABM. LAPF control always uses 7-bit sequence numbers; 3-bit sequence numbers are not allowed. The FCS for LAPF control is always the 16-bit CRC. Finally, the address field for LAPF control is two, three, or four octets long, containing a 10-bit, 16-bit, or 23-bit DLCI (data link connection identifier). The DLCI identifies a logical connection between a source and destination system. In addition, the address field contains some control bits that are useful for flow control purposes.

The LAPF core consists of the same flag, address, information, and FCS fields as LAPF control. The difference is that there is no control field for LAPF core. Thus, there is no means of doing flow and error control, which results in a more streamlined operation.

Asynchronous Transfer Mode (ATM)

Like frame relay, ATM is designed to provide a streamlined data-transfer capability across high-speed networks. Unlike frame relay, ATM is not based on HDLC. Instead, ATM is based on a completely new frame format, known as a cell, that provides minimum processing overhead.

The cell has a fixed length of 53 octets, or 424 bits. The details of the ATM cell fields are discussed in Part II.

6.6 RECOMMENDED READING

An excellent and very detailed treatment of flow control and error control is to be found in [BERT92]. A good survey of data link control protocols is [BLAC93].

BERT92 Bertsekas, D. and Gallager, R. *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.

BLAC93 Black, U. *Data Link Protocols*. Englewood Cliffs, NJ: Prentice Hall, 1993.

6.7 PROBLEMS

6.1 Consider a half-duplex point-to-point link using a stop-and-wait scheme.
   a. What is the effect on line utilization of increasing the message size so that fewer messages will be required? Other factors remain constant.
   b. What is the effect on line utilization of increasing the number of frames for a constant message size?
   c. What is the effect on line utilization of increasing frame size?

6.2 A channel has a data rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?

6.3 Consider the use of 1000-bit frames on a 1-Mbps satellite channel with a 270-ms delay. What is the maximum link utilization for
   a. Stop-and-wait flow control?
   b. Continuous flow control with a window size of 7?
   c. Continuous flow control with a window size of 127?
   d. Continuous flow control with a window size of 255?

6.4 In Figure 6.14, frames are generated at node *A* and sent to node *C* through node *B*. Determine the minimum transmission rate required between nodes *B* and *C* so that the buffers of node *B* are not flooded, based on the following:

- The data rate between *A* and *B* is 100 kbps.
- The propagation delay is 10 μsec/mile for both lines.
- There are full duplex lines between the nodes.
- All data frames are 1000 bits long; ACK frames are separate frames of negligible length.
- Between *A* and *B*, a sliding-window protocol with a window size of 3 is used.
- Between *B* and *C*, stop-and-wait is used.
- There are no errors.

Hint: In order not to flood the buffers of *B*, the average number of frames entering and leaving *B* must be the same over a long interval.

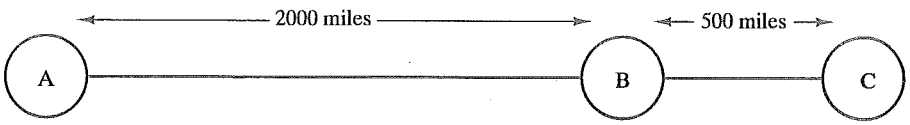


FIGURE 6.14 Configuration for problem 6.4

6.5 A channel has a data rate of *R* bps and a propagation delay of *t* seconds per kilometer. The distance between the sending and receiving nodes is *L* kilometers. Nodes exchange fixed-size frames of *B* bits. Find a formula that gives the minimum sequence field size of the frame as a function of *R*, *t*, *B*, and *L* (considering maximum utilization). Assume that ACK frames are negligible in size and the processing at the nodes is instantaneous.

6.6 Would you expect that the inclusion of a parity bit with each character would change the probability of receiving a correct message?

6.7 What is the purpose of using modulo 2 arithmetic rather than binary arithmetic in computing an FCS?

6.8 Consider a frame consisting of two characters of four bits each. Assume that the probability of bit error is 10^{-3} and that it is independent for each bit.
a. What is the probability that the received frame contains at least one error?
b. Now add a parity bit to each character. What is the probability?

6.9 Using the CRC-CCITT polynomial, generate the 16-bit CRC code for a message consisting of a 1 followed by 15 0s.
a. Use long division.
b. Use the shift register mechanism shown in Figure 6.6.

6.10 Explain in words why the shift register implementation of CRC will result in all 0s at the receiver if there are no errors. Demonstrate by example.

6.11 For $P = 110011$ and $M = 11100011$, find the CRC.

6.12 A CRC is constructed to generate a 4-bit FCS for an 11-bit message. The generator polynomial is $X^4 + X^3 + 1$.
a. Draw the shift register circuit that would perform this task (see Figure 6.6).
b. Encode the data bit sequence 10011011100 (leftmost bit is the least significant) using the generator polynomial and give the code word.
c. Now assume that bit 7 (counting from the LSB) in the code word is in error and show that the detection algorithm detects the error.

6.13 A modified CRC procedure is commonly used in communications standards. It is defined as follows:

$$\frac{X^{16}M(X) = X^k L(X)}{P(X)} = Q + \frac{R(X)}{P(X)}$$

$$\text{FCS} = L(X) + R(X)$$

where

$$L(X) = X^{15} + X^{14} + X^{13} + \ldots + X + 1$$

a. Describe in words the effect of this procedure.
b. Explain the potential benefits.

6.14 Why is it not necessary to have NAK0 and NAK1 for stop-and-wait ARQ?

6.15 Suppose that a selective-reject ARQ is used where $N = 4$. Show, by example, that a 3-bit sequence number is needed.

6.16 Using the same assumptions that are used for Figure 6.17 in Appendix 6A, plot line utilization as a function of P, the probability that a single frame is in error for the following error-control techniques:
a. Stop-and-wait.
b. Go-back-N with N = 7.
c. Go-back-N with N = 127.
d. Selective reject with N = 7.
e. Selective reject with N = 127.

Do all of the preçeding for the following values of a: 0.1, 1, 10, 100. Draw conclusions about which technique is appropriate for various ranges of a.

6.17 Two neighboring nodes (A and B) use a sliding-window protocol with a 3-bit sequence number. As the ARQ mechanism, Go-back-N is used with a window size of 4. Assuming A is transmitting and B is receiving, show the window positions for the following succession of events:
a. Before A sends any frames.
b. After A sends frames 0, 1, 2 and B acknowledges 0, 1 and the ACKs are received by A.
c. After A sends frames 3, 4, and 5 and B acknowledges 4 and the ACK is received by A.

6.18 It was stated in Section 6.3 that out-of-sequence acknowledgment could not be used for selective-reject ARQ. That is, if frame i is rejected by station X, all subsequent

I-frames and RR frames sent by X must have N(R) = i until frame i is successfully received, even if other frames with N(S) > i are successfully received in the meantime. One possible refinement is the following: N(R) = j in an I-frame or an RR frame is interpreted to mean that frame $j - 1$ and all preceding frames are accepted except for those that have been explicitly rejected using an SREJ frame. Comment on any possible drawback to this scheme.

6.19 The ISO standard for HDLC procedures (ISO 4335) includes the following definitions: (1) an REJ condition is considered cleared upon the receipt of an incoming I-frame with an N(S) equal to the N(R) of the outgoing REJ frame; and (2) an SREJ condition is considered cleared upon the receipt of an I-frame with an N(S) equal to the N(R) of the SREJ frame. The standard includes rules concerning the relationship between REJ and SREJ frames. These rules indicate what is allowable (in terms of transmitting REJ and SREJ frames) if an REJ condition has not yet been cleared and what is allowable if an SREJ condition has not yet been cleared. Deduce the rules and justify your answer.

6.20 Two stations communicate via a 1-Mbps satellite link with a propagation delay of 270 ms. The satellite serves merely to retransmit data received from one station to another, with negligible switching delay. Using HDLC frames of 1024 bits with 3-bit sequence numbers, what is the maximum possible data throughput (not counting overhead bits)?

6.21 It is clear that bit stuffing is needed for the address, data, and FCS fields of an HDLC frame. Is it needed for the control field?

6.22 Suggest improvements to the bit stuffing-algorithm to overcome the problems of single-bit errors.

6.23 Using the example bit string of Figure 6.11, show the signal pattern on the line using NRZ-L coding; does this suggest a side benefit of bit stuffing?

6.24 Assume that the primary HDLC station in NRM has sent six I-frames to a secondary. The primary's N(S) count was three (011 binary) prior to sending the six frames. If the poll bit is on in the sixth frame, what will be the N(R) count back from the secondary after the last frame? Assume error-free operation.

6.25 Consider that several physical links connect two stations. We would like to use a "multilink HDLC" that makes efficient use of these links by sending frames on an FIFO basis on the next available link. What enhancements to HDLC are needed?

6A APPENDIX

PERFORMANCE ISSUES

IN THIS APPENDIX, we examine some of the performance issues related to the use of sliding-window flow-control.

Stop-and-Wait Flow Control

Let us determine the maximum potential efficiency of a half-duplex point-to-point line using the stop-and-wait scheme described in Section 6.1. Suppose that a long message is to be sent as a sequence of frames $f_1, f_2, \ldots f_n$, in the following fashion:

- Station S_1 sends f_1.
- Station S_2 sends an acknowledgment.
- Station S_1 sends f_2.
- Station S_2 sends an acknowledgment.
- .
- .
- .
- Station S_1 sends f_n.
- Station S_2 sends an acknowledgment.

The total time to send the data, T, can be expressed as $T = nT_F$, where T_F is the time to send one frame and receive an acknowledgment. We can express T_F as follows:

$$T_F = t_{\text{prop}} + t_{\text{frame}} + t_{\text{proc}} + t_{\text{prop}} + t_{\text{ack}} + t_{\text{proc}}$$

where

t_{prop} = propagation time from S_1 to S_2
t_{frame} = time to transmit a frame (time for the transmitter to send out all of the bits of the frame)
t_{ack} = processing time at each station to react to an incoming event
t_{proc} = time to transmit an acknowledgment

Let us assume that the processing time is relatively negligible, and that the acknowledgment frame is very small compared to a data frame, both of which are reasonable assumptions. Then we can express the total time to send the data as

$$T = n(2t_{\text{prop}} + t_{\text{frame}})$$

Of that time, only $n \times t_{\text{frame}}$ is actually spent transmitting data and the rest is overhead. The utilization, or efficiency, of the line is

$$U = \frac{n \times t_{\text{frame}}}{n(2t_{\text{prop}} + t_{\text{frame}})}$$

$$= \frac{t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}}$$

It is useful to define the parameter $a = t_{\text{prop}}/t_{\text{frame}}$. Then,

$$U = \frac{1}{1 + 2a} \tag{6.2}$$

This is the maximum possible utilization of the link. Because the frame contains overhead bits, actual utilization is lower. The parameter a is constant if both t_{prop} and t_{frame} are constants, which is typically the case. Fixed length frames are often used for all except the last frame in a sequence, and the propagation delay is constant for point-to-point links.

To get some insight into Equation (6.2), let us derive a different expression for a. We have

$$a = \frac{\text{Propagation Time}}{\text{Transmission Time}} \tag{6.3}$$

The propagation time is equal to the distance d of the link divided by the velocity of propagation V. For unguided transmission through air or space, V is the speed of light, 3×10^8 m/sec. For guided transmission, V is approximately the speed of light for optical fiber and about 0.67 times the speed of light for copper media. The transmission time is equal to the length of the frame in bits, L, divided by the data rate R. Therefore,

$$a = \frac{d/V}{L/R} = \frac{Rd}{VL}$$

Thus, for fixed-length frames and a fixed distance between stations, a is proportional to the data rate times the length of the medium. A useful way of looking at a is that it represents the length of the medium in bits ($R \times {}^{d}/_{V}$) compared to the frame length (L).

With this interpretation in mind, Figure 6.2 illustrates equation (6.2). In this figure, transmission time is normalized to 1 and, hence, the propagation time, by Equation (6.3), is a. For the case of $a < 1$, the link's bit length is less than that of the frame. The station T begins transmitting a frame at time t_0. At $t_0 + a$, the leading edge of the frame reaches the receiving station R, while T is still in the process of transmitting the frame. At $t_0 + 1$, T completes transmission. At $t_0 + 1 + a$, R has received the entire frame and immediately transmits a small acknowledgment frame. This acknowledgment arrives back at T at $t_0 + 1 + 2a$. Total elapsed time: $1 + 2a$. Total transmission time: 1. Hence, utilization is $1/(1 + 2a)$. The same result is achieved with a > 1, as illustrated in Figure 6.2.

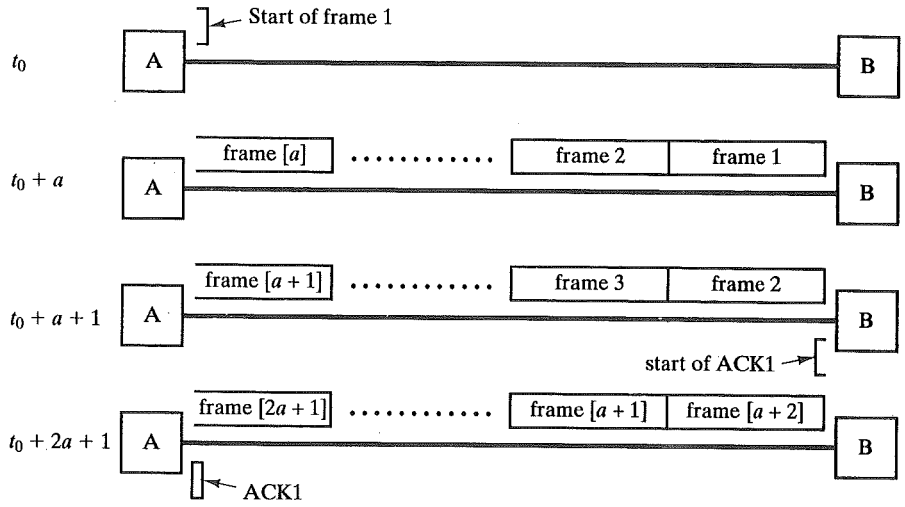
Let us consider a few examples. First, consider a wide-area network (WAN) using ATM (asynchronous transfer mode, described in Part II), with the two stations a thousand kilometers apart. The standard ATM frame size (called a cell) is 424 bits and one of the standardized data rates is 155.52 Mbps. Thus, transmission time equals $424/(155.52 \times 10^6) = 2.7 \times 10^{-6}$ seconds. If we assume an optical fiber link, then the propagation time is $(10^6 \text{ meters})/(3 \times 10^8 \text{ m/sec}) = 0.33 \times 10^{-2}$ seconds. Thus, $a = (0.33 \times 10^{-2})/(2.7 \times 10^{-6}) = 1200$, and efficiency is only $1/1401 = 0.0007$!

At the other extreme, in terms of distance, is the local area network (LAN). Distances range from 0.1 to 10 km, with data rates of 10 to 100 Mbps; higher data rates tend to be associated with shorter distances. Using a value of $V = 2 \times 10^8$ m/sec, a frame size of 1000 bits, and a data rate of 10 Mbps, the value of a is in the range of 0.005 to 0.5; this yields a utilization in the range of 0.5 to 0.99. For a 100-Mbps LAN, given the shorter distances, comparable utilizations are possible.

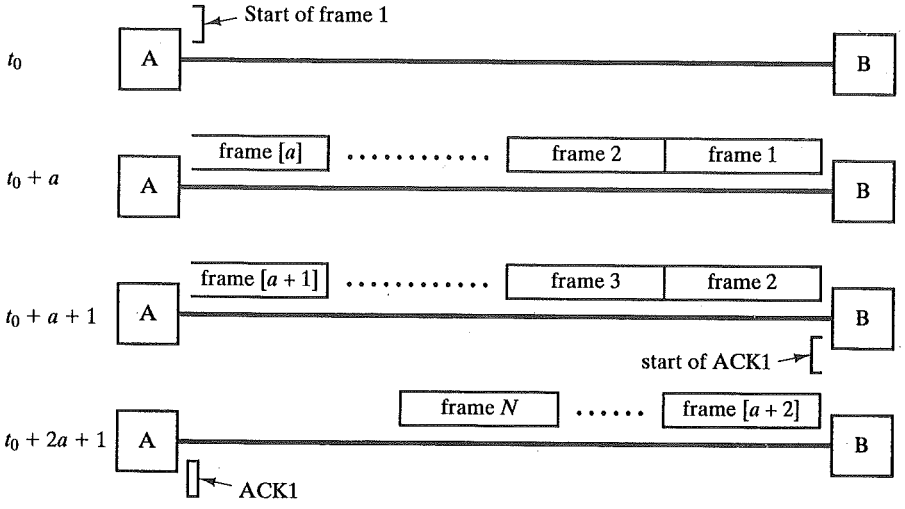
We can see that LANs are typically quite efficient, whereas high-speed WANs are not. As a final example, let us consider digital data transmission via modem over a voice-grade line. A practical upper bound on data rate is 28.8 kbps. Again, let us consider a 1000-bit frame. The link distance can be anywhere from a few tens of meters to thousands of kilometers. If we pick, say, as a short distance, $d = 1000$ m, then $a = (28{,}800 \text{ bps} \times 1000 \text{ m})/(2 \times 10^8 \text{ m/sec} \times 1000 \text{ bits}) = 1.44 \times 10^{-4}$, and utilization is effectively 1.0. Even in a long-distance case, such as $d = 5000$ km, we have $a = (28{,}800 \times 5 \times 10^6)/(2 \times 10^8 \times 1000 \text{ bits}) = 0.72$ and efficiency equals 0.4.

Sliding-Window Control

For sliding-window flow control, the efficiency of the line depends on both the window size, N, and the value of a. For convenience, let us again normalize frame transmission time to a value of 1; thus, the propagation time is a. Figure 6.15 illustrates the efficiency of a full-duplex point-to-point line. Station A begins to emit a sequence of frames at time t_0. The leading edge



(a) $N > 2a + 1$



(b) $N < 2a + 1$

[X] = smallest integer greater than or equal to [X]

FIGURE 6.15 Timing of a sliding-window protocol.

of the first frame reaches station B at $t_0 + a$. The first frame is entirely absorbed by $t_0 + a + 1$. Assuming negligible processing time, B can immediately acknowledge the first frame (ACK1). Let us also assume that the acknowledgment frame is so small that transmission time is negligible. Then the ACK1 reaches A at $t_0 + 2a + 1$. To evaluate performance, we need to consider two cases:

- Case 1: $N > 2a + 1$. The acknowledgment for frame 1 reaches A before A has exhausted its window. Thus, A can transmit continuously with no pause, and utilization is 1.0.
- Case 2: $N < 2a + 1$. A exhausts its window at $t_0 + N$ and cannot send additional frames until $t_0 + 2a + 1$. Thus, line utilization is N time units out of a period of $(2a + 1)$ time units.

Therefore, we can state that

$$U = \begin{cases} 1 & N > 2a + 1 \\ \dfrac{N}{2a + 1} & N < 2a + 1 \end{cases} \tag{6.4}$$

Typically, the sequence number is provided for in an n-bit field, and the maximum window size is $N = 2^n - 1$ (not 2^n; this is explained in Section 6.3). Figure 6.16 shows the maximum efficiency achievable for window sizes of 1, 7, and 127 as a function of a. A window size of 1 corresponds to stop-and-wait. A window size of 7 (3 bits) is adequate for many applications. A window size of 127 (7 bits) is adequate for larger values of a, such as may be found in high-speed WANs.

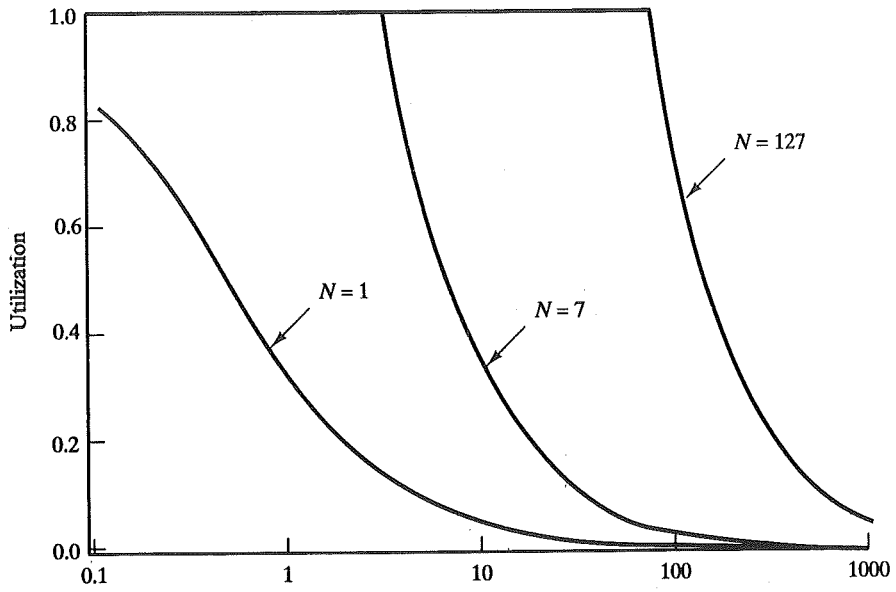


FIGURE 6.16 Line utilization as a function of window size.

ARQ

We have seen that sliding-window flow control is more efficient than stop-and-wait flow control. We would expect that when error-control functions are added, this would still be true—that is, that go-back-N and selective-reject ARQ are more efficient than stop-and-wait ARQ. Let us develop some approximations to determine the degree of improvement to be expected.

First, consider stop-and-wait ARQ. With no errors, the maximum utilization is $1/(1 + 2a)$ as shown in Equation (6.2). We want to account for the possibility that some frames are repeated because of bit errors. To start, note that the utilization U can be defined as

$$U = \frac{T_f}{T_t} \tag{6.5}$$

where

T_f = time for transmitter to emit a single frame
Tt = total time that line is engaged in the transmission of a single frame

For error-free operation using stop-and-wait ARQ,

$$U = \frac{T_f}{T_f + 2T_p}$$

where T_p is the propagation time. Dividing by T_f and remembering that $a = T_p/T_f$, we again have Equation (6.2). If errors occur, we must modify Equation (6.5) to

$$U = \frac{T_f}{N_r T_t}$$

where N_r is the expected number of transmissions of a frame. Thus, for stop-and-wait ARQ, we have

$$U = \frac{1}{N_r(1 + 2a)}$$

A simple expression for N_r can be derived by considering the probability P that a single frame is in error. If we assume that ACKs and NAKs are never in error, the probability that it will take exactly k attempts to transmit a frame successfully is $P^{k-1}(1 - P)$. That is, we have (k – 1) unsuccessful attempts followed by one successful attempt; the probability of this occurring is just the product of the probability of the individual events occurring. Then,[1]

$$N_r = \text{E[transmissions]} = \sum_{i=1}^{\infty} (i \times P_r[i \text{ transmissions}])$$

$$= \sum_{i=1}^{\infty} (iP^{i-1}(1 - P)) = \frac{1}{1 - P}$$

So we have

$$\textit{Stop-and-Wait:} \qquad U = \frac{1 - P}{1 + 2a}$$

[1] This derivation uses the equality $\sum_{i=1}^{\infty} (iX^{i-1}) = \frac{1}{(1 - X)^2}$ for $(-1 < X < 1)$

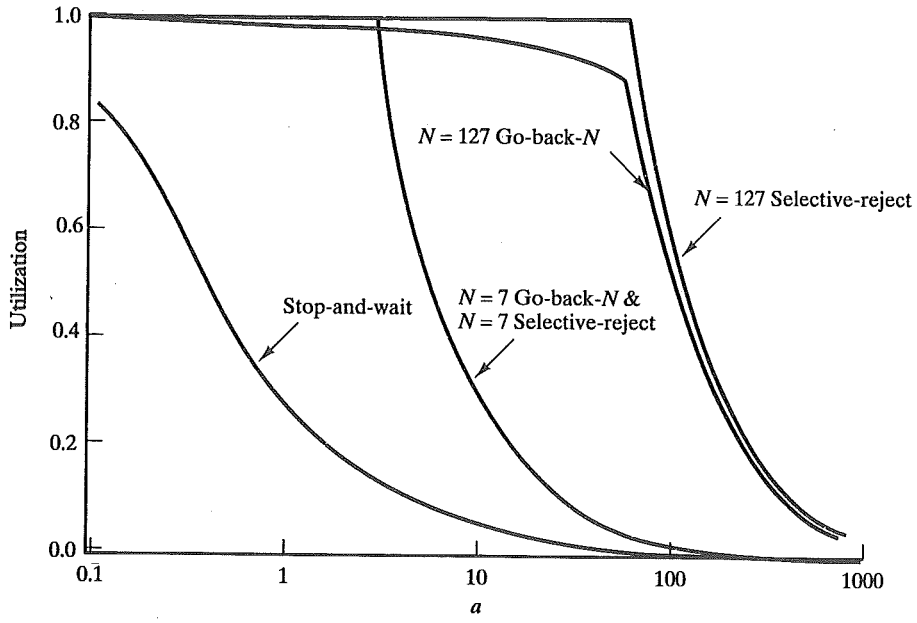


FIGURE 6.17 Line utilization for various error-control techniques ($P = 10^{-3}$).

For the sliding-window protocol, Equation (6.4) applies for error-free operation. For selective-reject ARQ, we can use the same reasoning as applied to stop-and-wait ARQ. That is, the error-free equations must be divided by N_r. Again, $N_r = 1/(1 - P)$. So,

$$\textit{Selective reject:} \qquad U = \begin{cases} 1 - P & N > 2a + 1 \\ \dfrac{N(1 - P)}{2a + 1} & N < 2a + 1 \end{cases}$$

The same reasoning applies for go-back-N ARQ, but we must be more careful in approximating N_r. Each error generates a requirement to retransmit K frames rather than just one frame. Thus,

$$\begin{aligned} N_r &= E[\textit{number of transmitted frames to successfully transmit one frame}] \\ &= \sum_{i=1}^{\infty} f(i)P^{i-1}(1 - P) \end{aligned}$$

where $f(i)$ is the total number of frames transmitted if the original frame must be transmitted i times. This can be expresses as

$$\begin{aligned} f(i) &= 1 + (i - 1)K \\ &= (1 - K) + Ki \end{aligned}$$

Substituting yields[2]

[2] This derivation uses the equality $\sum_{i=1}^{\infty} X^{i-1} = \dfrac{1}{1 - X}$ for $(-1 < X < 1)$.

$$N_r = (1 - K)\sum_{i=1}^{\infty} P^{i-1}(1 - P) + K\sum_{i=1}^{\infty} iP^{i-1}(1 - P)$$

$$= 1 - K + \frac{K}{1 - P}$$

$$= \frac{1 - P + KP}{1 - P}$$

By studying Figure 6.15, the reader should conclude that K is approximately equal to $(2a + 1)$ for $N > (2a + 1)$, and $K = N$ for $N < (2a + 1)$. Thus,

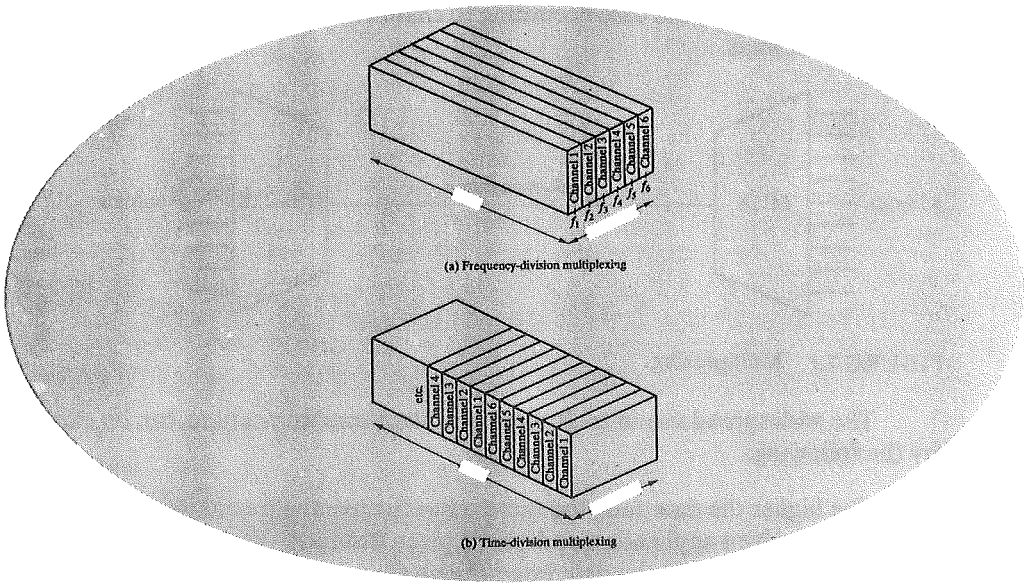
Go-back-N:
$$U = \begin{cases} \dfrac{1 - P}{1 + 2aP} & N > 2a + 1 \\ \dfrac{N(1 - P)}{(2a + 1)(1 - P + NP)} & N < 2a + 1 \end{cases}$$

Note that for $N = 1$, both selective-reject and go-back-N ARQ reduce to stop-and-wait. Figure 6.17 compares these three error-control techniques for a value of P = 10 − 3.[3] This figure and the equations are only approximations. For example, we have ignored errors in acknowledgment frames and, in the case of go-back-N, we have also ignored errors in retransmitted frames other than the frame initially in error. However, the results do give an indication of the relative performance of the three techniques.

[3] For $N = 7$, the curves for go-back-N and selective-reject are so close that they appear to be identical in the figure.

CHAPTER 7

MULTIPLEXING



In Chapter 6, we described efficient techniques for utilizing a data link under heavy load. Specifically, with two devices connected by a point-to-point link, it is generally desirable to have multiple frames outstanding so that the data link does not become a bottleneck between the stations. Now consider the opposite problem. Typically, two communicating stations will not utilize the full capacity of a data link. For efficiency, it should be possible to share that capacity. A generic term for such sharing is multiplexing.

A common application of multiplexing is in long-haul communications. Trunks on long-haul networks are high-capacity fiber, coaxial, or microwave links. These links can carry large numbers of voice and data transmissions simultaneously using multiplexing.

Figure 7.1 depicts the multiplexing function in its simplest form. There are n inputs to a multiplexer. The multiplexer is connected by a single data link to a demultiplexer. The link is able to carry n separate channels of data. The multiplexer combines (multiplexes) data from the n input lines and transmits over a higher-capacity data link. The demultiplexer accepts the multiplexed data stream, separates (demultiplexes) the data according to channel, and delivers them to the appropriate output lines.

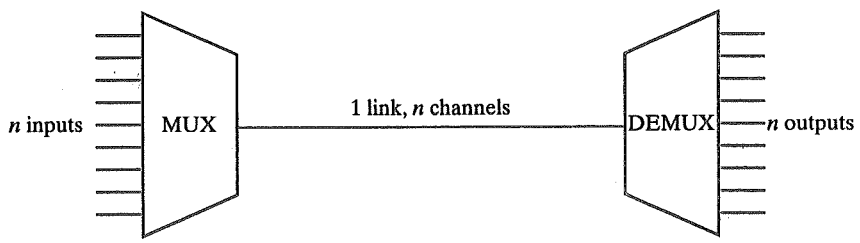


FIGURE 7.1 Multiplexing.

The widespread use of multiplexing in data communications can be explained by the following:

1. The higher the data rate, the more cost-effective the transmission facility. That is, for a given application and over a given distance, the cost per kbps declines with an increase in the data rate of the transmission facility. Similarly, the cost of transmission and receiving equipment, per kbps, declines with increasing data rate.
2. Most individual data-communicating devices require relatively modest data-rate support. For example, for most terminal and personal computer applications, a data rate of between 9600 bps and 64 kbps is generally adequate.

The preceding statements were phrased in terms of data communicating devices. Similar statements apply to voice communications; that is, the greater the capacity of a transmission facility, in terms of voice channels, also, the less the cost per individual voice channel; so, the capacity required for a single voice channel is modest.

This chapter concentrates on three types of multiplexing techniques. The first, frequency-division multiplexing (FDM), is the most heavily used and is familiar to

anyone who has ever turned on a radio or television set. The second is a particular case of time-division multiplexing (TDM) known as synchronous TDM. This is commonly used for multiplexing digitized voice streams and data streams. The third type seeks to improve on the efficiency of synchronous TDM by adding complexity to the multiplexer. It is known by a variety of names, including statistical TDM, asynchronous TDM, and intelligent TDM. This book uses the term statistical TDM, which highlights one of its chief properties.

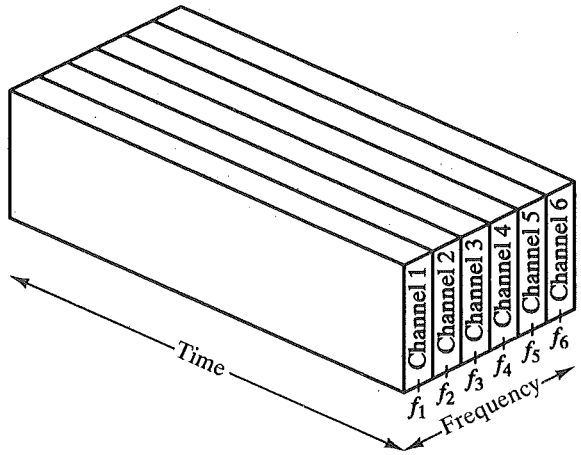
7.1 FREQUENCY-DIVISION MULTIPLEXING

Characteristics

FDM is possible when the useful bandwidth of the transmission medium exceeds the required bandwidth of signals to be transmitted. A number of signals can be carried simultaneously if each signal is modulated onto a different carrier frequency and the carrier frequencies are sufficiently separated that the bandwidths of the signals do not overlap. A general case of FDM is shown in Figure 7.2a. Six signal sources are fed into a multiplexer, which modulates each signal onto a different frequency ($f_1, \ldots, f_6$). Each modulated signal requires a certain bandwidth centered around its carrier frequency, referred to as a *channel*. To prevent interference, the channels are separated by guard bands, which are unused portions of the spectrum.

The composite signal transmitted across the medium is analog. Note, however, that the input signals may be either digital or analog. In the case of digital input, the input signals must be passed through modems to be converted to analog. In either case, each input analog signal must then be modulated to move it to the appropriate frequency band.

A familiar example of FDM is broadcast and cable television. The television signal discussed in Chapter 2 fits comfortably into a 6-MHz bandwidth. Figure 7.3 depicts the transmitted TV signal and its bandwidth. The black-and-white video signal is AM modulated on a carrier signal f_{cv}. Because the baseband video signal has a bandwidth of 4 MHz, we would expect the modulated signal to have a bandwidth of 8 MHz centered on f_{cv}. To conserve bandwidth, the signal is passed through a sideband filter so that most of the lower sideband is suppressed. The resulting signal extends from about $f_{cv} - 0.75$ MHz to $f_{cv} + 4.2$ MHz. A separate color subcarrier, f_{cc}, is used to transmit color information. This is spaced far enough from f_{cv} that there is essentially no interference. Finally, the audio portion of the signal is modulated on f_{ca}, outside the effective bandwidth of the other two signals. A bandwidth of 50 kHz is allocated for the audio signal. The composite signal fits into a 6-MHz bandwidth with the video, color, and audio signal carriers at 1.25 MHz, 4.799545 MHz, and 5.75 MHz, respectively, above the lower edge of the band. Thus, multiple TV signals can be frequency-division multiplexed on a CATV cable, each with a bandwidth of 6 MHz. Given the enormous bandwidth of coaxial cable (as much as 500 MHz), dozens of TV signals can be simultaneously carried using FDM. Of course, using radio-frequency propagation through the atmosphere is also a form of FDM; Table 7.1 shows the frequency allocation in the United States for broadcast television.



(a) Frequency-division multiplexing

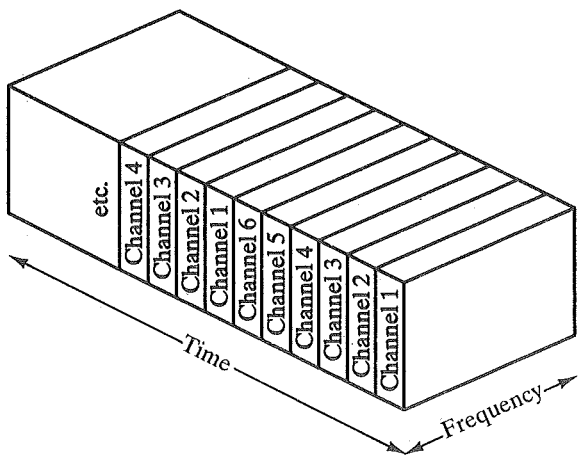
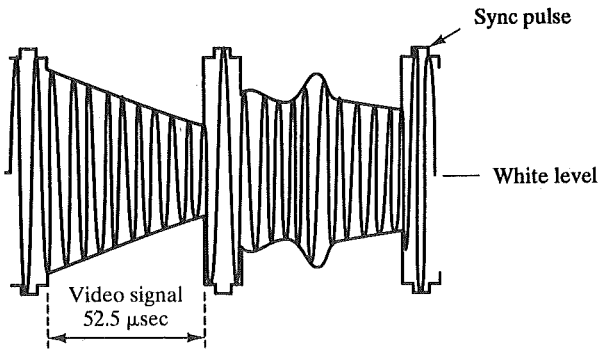


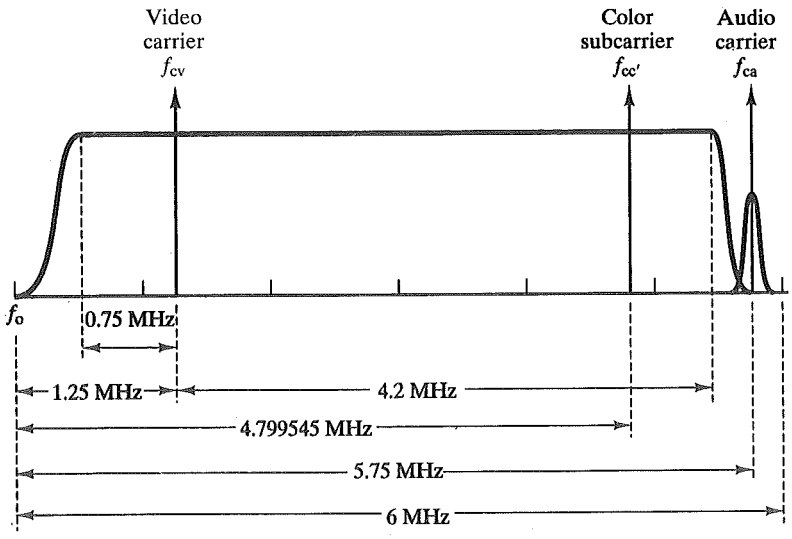
FIGURE 7.2 FDM and TDM.

A generic depiction of an FDM system is shown in Figure 7.4. A number of analog or digital signals $[m_i(t), i = 1, N]$ are to be multiplexed onto the same transmission medium. Each signal $m_i(t)$ is modulated onto a carrier f_{sci}; because multiple carriers are to be used, each is referred to as a subcarrier. Any type of modulation may be used. The resulting modulated analog signals are then summed to produce a composite signal $m_c(t)$. Figure 7.4b shows the result. The spectrum of signal $m_i(t)$ is shifted to be centered on f_{sci}. For this scheme to work, f_{sci} must be chosen so that the bandwidths of the various signals do not overlap; otherwise, it will be impossible to recover the original signals.

The composite signal may then be shifted as a whole to another carrier frequency by an additional modulation step. We will see examples of this below. This second modulation step need not use the same modulation technique as the first.



(a) Amplitude modulation with video signal



(b) Magnitude spectrum of RF video signal

FIGURE 7.3 Transmitted TV signal.

The composite signal has a total bandwidth B, where $B > \sum_{i=1}^{N} B_{sci}$. This analog signal may be transmitted over a suitable medium. At the receiving end, the composite signal is passed through N bandpass filters, each filter centered on f_{sci} and having a bandwidth B_{sci}, for $1 < i < N$; in this way, the signal is again split into its component parts. Each component is then demodulated to recover the original signal.

Let us consider a simple example of transmitting three voice signals simultaneously over a medium. As was mentioned, the bandwidth of a voice signal is generally taken to be 4 kHz, with an effective spectrum of 300 to 3400 Hz (Figure 7.5a). If such a signal is used to amplitude-modulate a 64-kHz carrier, the spectrum of Fig-

TABLE 7.1 Broadcast television channel frequency allocation.

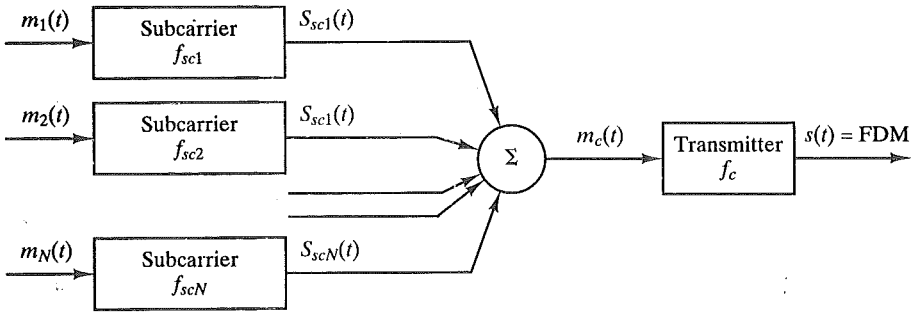
Channel number	Band (MHz)	Channel number	Band (MHz)	Channel number	Band (MHz)
2	54–60	25	536–542	48	674–680
3	60–66	26	542–548	49	680–686
4	66–72	27	548–554	50	686–692
5	76–82	28	554–560	51	692–698
6	82–88	29	560–566	52	698–704
7	174–180	30	566–572	53	704–710
8	180–186	31	572–578	54	710–716
9	186–192	32	578–584	55	716–722
10	192–198	33	584–590	56	722–728
11	198–204	34	590–596	57	728–734
12	204–210	35	596–602	58	734–740
13	210–216	36	602–608	59	740–746
14	470–476	37	608–614	60	746–752
15	476–482	38	614–620	61	752–758
16	482–488	39	620–626	62	758–764
17	488–494	40	626–632	63	764–770
18	494–500	41	632–638	64	770–776
19	500–506	42	638–644	65	776–782
20	506–512	43	644–650	66	782–788
21	512–518	44	650–656	67	788–794
22	518–524	45	656–662	68	794–800
23	524–530	46	662–668	69	800–806
24	530–536	47	668–674		

ure 7.5b results. The modulated signal has a bandwidth of 8 kHz, extending from 60 to 68 kHz. To make efficient use of bandwidth, we elect to transmit only the lower sideband. Now, if three voice signals are used to modulate carriers at 64, 68, and 72 kHz, and only the lower sideband of each is taken, the spectrum of Figure 7.5c results.

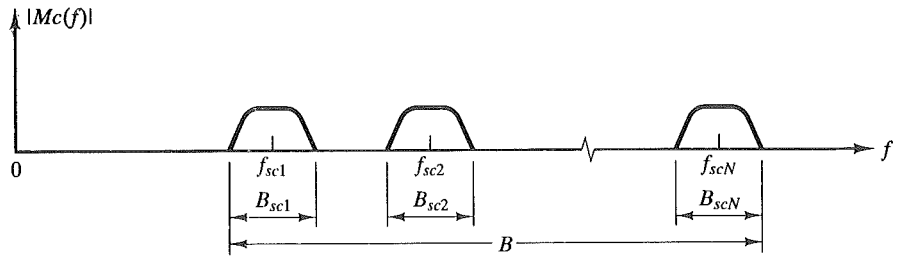
This figure points out two problems that an FDM system must cope with. The first is crosstalk, which may occur if the spectra of adjacent component signals overlap significantly. In the case of voice signals, with an effective bandwidth of only 3100 Hz (300 to 3400), a 4-kHz bandwidth is adequate. The spectra of signals produced by modems for voiceband transmission also fit well in this bandwidth. Another potential problem is intermodulation noise, which was discussed in Chapter 2. On a long link, the nonlinear effects of amplifiers on a signal in one channel could produce frequency components in other channels.

Analog Carrier Systems

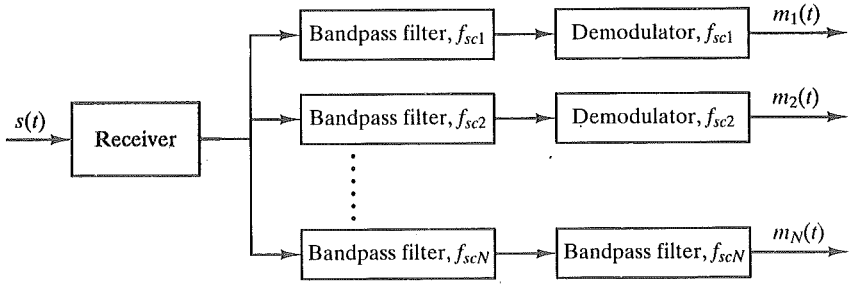
The long-distance carrier system provided in the United States and throughout the world is designed to transmit voiceband signals over high-capacity transmission links, such as coaxial cable and microwave systems. The earliest, and still most common, technique for utilizing high-capacity links is FDM. In the United States, AT&T has designated a hierarchy of FDM schemes to accommodate transmission systems of various capacities. A similar, but unfortunately not identical, system has been adopted internationally under the auspices of ITU-T (Table 7.2).



(a) Transmitter



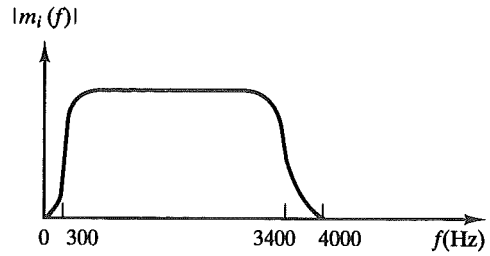
(b) Spectrum of composite signal (positive f)



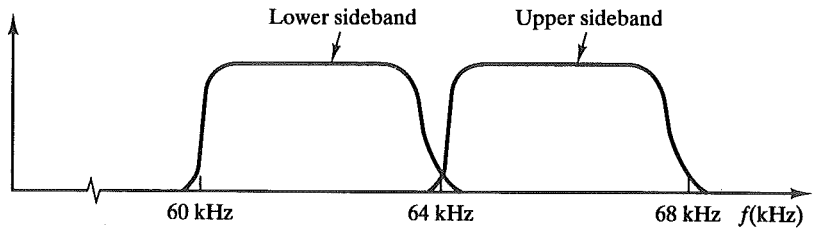
(c) Receiver

FIGURE 7.4 Frequency division multiplexing.

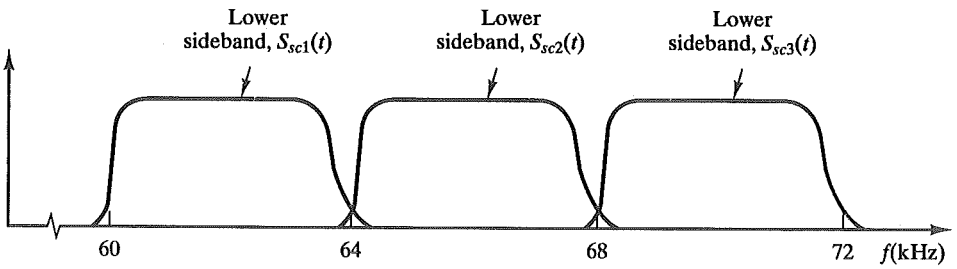
At the first level of the AT&T hierarchy, 12 voice channels are combined to produce a group signal with a bandwidth of 12 × 4 kHz = 48 kHz, in the range 60 to 108 kHz. The signals are produced in a fashion similar to that described above, using subcarrier frequencies of from 64 to 108 kHz in increments of 4 kHz. The next basic building block is the 60-channel supergroup, which is formed by frequency-division multiplexing five-group signals. At this step, each group is treated as a single signal with a 48-kHz bandwidth and is modulated by a subcarrier. The subcarriers have frequencies from 420 to 612 kHz in increments of 48 kHz. The resulting signal occupies 312 to 552 kHz.



(a) Spectrum of $m_i(t)$, positive f



(b) Spectrum of $S_{sc1}(t)$ for $f_{sc1} = 64$ kHz



(c) Spectrum of composite signal using subcarriers at 64 kHz, 68 kHz, and 72 kHz

FIGURE 7.5 FDM of three voiceband signals.

There are several variations to supergroup formation. Each of the five inputs to the supergroup multiplexer may be a group channel containing 12 multiplexed voice signals. In addition, any signal up to 48 kHz wide whose bandwidth is contained within 60 to 108 kHz may be used as input to the supergroup multiplexer. As another variation, it is possible to directly combine 60 voiceband channels into a supergroup; this may reduce multiplex costs where an interface with existing-group multiplex is not required.

The next level of the hierarchy is the mastergroup that combines 10 supergroup inputs. Again, any signal with a bandwidth of 240 kHz in the range 312 to 552 kHz can serve as input to the mastergroup multiplexer. The mastergroup has a bandwidth of 2.52 MHz and can support 600 voice-frequency (VF) channels. Higher-level multiplexing is defined above the mastergroup, as shown in Table 7.2.

Note that the original voice or data signal may be modulated many times. For example, a data signal may be encoded using QPSK to form an analog voice signal.

TABLE 7.2 North American and international FDM carrier standards.

Number of voice channels	Bandwidth	Spectrum	AT&T	ITU-T
12	48 kHz	60–108 kHz	Group	Group
60	240 kHz	312–552 kHz	Supergroup	Supergroup
300	1.232 MHz	812–2044 kHz		Mastergroup
600	2.52 MHz	564–3084 kHz	Mastergroup	
900	3.872 MHz	8.516–12.388 MHz		Supermaster group
$N \times 600$			Mastergroup multiplex	
3,600	16.984 MHz	0.564–17.548 MHz	Jumbogroup	
10,800	57.442 MHz	3.124–60.566 MHz	Jumbogroup multiplex	

This signal could then be used to modulate a 76-kHz carrier to form a component of a group signal. This group signal could then be used to modulate a 516-kHz carrier to form a component of a supergroup signal. Each stage can distort the original data; this is so, for example, if the modulator/multiplexer contains nonlinearities or if it introduces noise.

7.2 SYNCHRONOUS TIME-DIVISION MULTIPLEXING

Characteristics

Synchronous time-division multiplexing is possible when the achievable data rate (sometimes, unfortunately, called bandwidth) of the medium exceeds the data rate of digital signals to be transmitted. Multiple digital signals (or analog signals carrying digital data) can be carried on a single transmission path by interleaving portions of each signal in time. The interleaving can be at the bit level or in blocks of bytes or larger quantities. For example, the multiplexer in Figure 7.2b has six inputs which might each be, say, 9.6 kbps. A single line with a capacity of at least 57.6 kbps (plus overhead capacity) could accommodate all six sources.

A generic depiction of a synchronous TDM system is provided in Figure 7.6. A number of signals $[m_i(t), i = 1, N]$ are to be multiplexed onto the same transmission medium. The signals carry digital data and are generally digital signals. The incoming data from each source are briefly buffered. Each buffer is typically one bit or one character in length. The buffers are scanned sequentially to form a composite digital data stream $m_c(t)$. The scan operation is sufficiently rapid so that each buffer is emptied before more data can arrive. Thus, the data rate of $m_c(t)$ must at least equal the sum of the data rates of the $m_i(t)$. The digital signal $m_c(t)$ may be transmitted directly or passed through a modem so that an analog signal is transmitted. In either case, transmission is typically synchronous.

The transmitted data may have a format something like Figure 7.6b. The data are organized into frames. Each frame contains a cycle of time slots. In each frame,

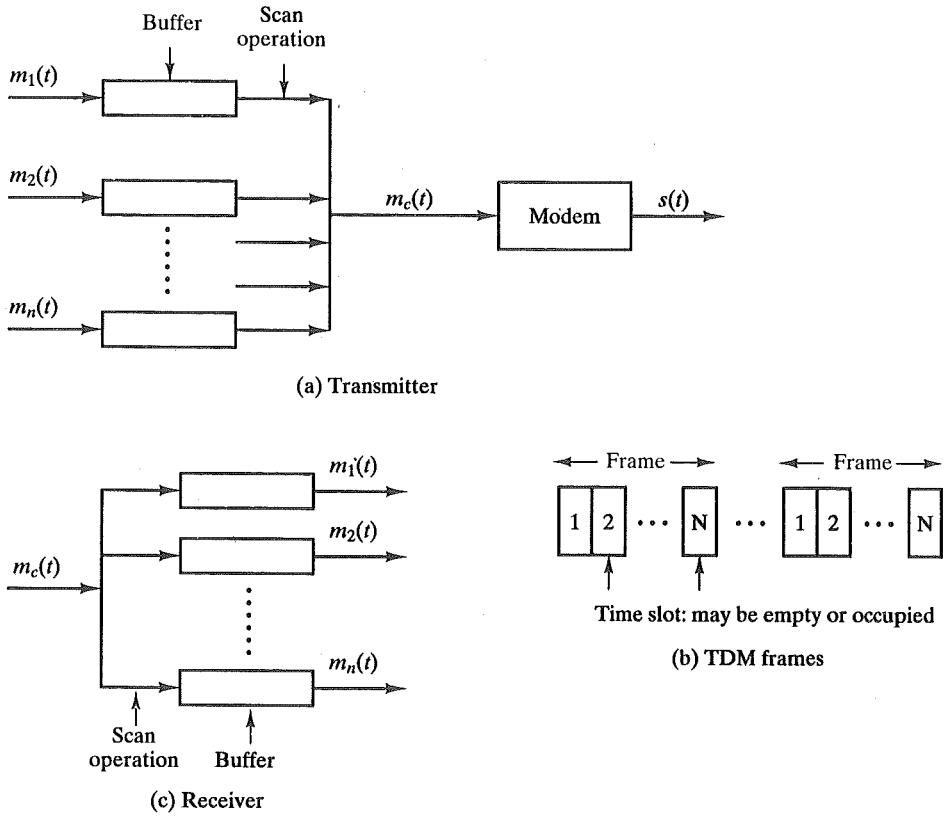


FIGURE 7.6 Synchronous time-division multiplexing.

one or more slots is dedicated to each data source. The sequence of slots dedicated to one source, from frame to frame, is called a channel. The slot length equals the transmitter buffer length, typically a bit or a character.

The character-interleaving technique is used with asynchronous sources. Each time slot contains one character of data. Typically, the start and stop bits of each character are eliminated before transmission and reinserted by the receiver, thus improving efficiency. The bit-interleaving technique is used with synchronous sources and may also be used with asynchronous sources. Each time slot contains just one bit.

At the receiver, the interleaved data are demultiplexed and routed to the appropriate destination buffer. For each input source $m_i(t)$, there is an identical output source which will receive the input data at the same rate at which it was generated.

Synchronous TDM is called synchronous not because synchronous transmission is used, but because the time slots are preassigned to sources and fixed. The time slots for each source are transmitted whether or not the source has data to send; this is, of course, also the case with FDM. In both cases, capacity is wasted to achieve simplicity of implementation. Even when fixed assignment is used, however, it is possible for a synchronous TDM device to handle sources of different data

rates. For example, the slowest input device could be assigned one slot per cycle, while faster devices are assigned multiple slots per cycle.

TDM Link Control

The reader will note that the transmitted data stream depicted in Figure 7.6 does not contain the headers and trailers that we have come to associate with synchronous transmission. The reason is that the control mechanisms provided by a data link protocol are not needed. It is instructive to ponder this point, and we do so by considering two key data link control mechanisms: flow control and error control. It should be clear that, as far as the multiplexer and demultiplexer (Figure 7.1) are concerned, flow control is not needed. The data rate on the multiplexed line is fixed, and the multiplexer and demultiplexer are designed to operate at that rate. But suppose that one of the individual output lines attaches to a device that is temporarily unable to accept data? Should the transmission of TDM frames cease? Clearly not, as the remaining output lines are expecting to receive data at predetermined times. The solution is for the saturated output device to cause the flow of data from the corresponding input device to cease. Thus, for a while, the channel in question will carry empty slots, but the frames as a whole will maintain the same transmission rate.

The reasoning for error control is the same. It would not do to request retransmission of an entire TDM frame because an error occurs on one channel. The devices using the other channels do not want a retransmission nor would they know that a retransmission has been requested by some other device on another channel. Again, the solution is to apply error control on a per-channel basis.

How are flow control, error control, and other good things to be provided on a per-channel basis? The answer is simple: Use a data link control protocol such as HDLC on a per-channel basis. A simplified example is shown in Figure 7.7. We

(a) Configuration

Input$_1$ ..... F_1 f_1 f_1 d_1 d_1 d_1 C_1 A_1 F_1 f_1 f_1 d_1 d_1 d_1 C_1 A_1 F_1

Input$_2$ ... F_2 f_2 f_2 d_2 d_2 d_2 d_2 C_2 A_2 F_2 f_2 f_2 d_2 d_2 d_2 d_2 C_2 A_2 F_2

(b) Input data stream

... f_2 F_1 d_2 f_1 d_2 f_1 d_2 d_1 d_2 d_1 C_2 d_1 A_2 C_1 F_2 A_1 f_2 F_1 f_2 f_1 d_2 f_1 d_2 d_1 d_2 d_1 d_2 d_1 C_2 C_1 A_2 A_1 F_2 F_1

(c) Multiplexed data stream

LEGEND

F = flag field
A = address field
C = control field
d = one octet of data field
f = one octet of FCS field

FIGURE 7.7 Use of data link control on TDM channels.

assume two data sources, each using HDLC. One is transmitting a stream of HDLC frames containing three octets of data; the other is transmitting HDLC frames containing four octets of data. For clarity, we assume that character-interleaved multiplexing is used, although bit interleaving is more typical. Notice what is happening. The octets of the HDLC frames from the two sources are shuffled together for transmission over the multiplexed line. The reader may initially be uncomfortable with this diagram, as the HDLC frames have lost their integrity in some sense. For example, each frame check sequence (FCS) on the line applies to a disjointed set of bits. Even the FCS is not in one piece! However, the pieces are reassembled correctly before they are seen by the device on the other end of the HDLC protocol. In this sense, the multiplexing/demultiplexing operation is transparent to the attached stations; to each communicating pair of stations, it appears that they have a dedicated link.

One refinement is needed in Figure 7.7. Both ends of the line need to be a combination multiplexer/demultiplexer with a full-duplex line in between. Then each channel consists of two sets of slots, one traveling in each direction. The individual devices attached at each end can, in pairs, use HDLC to control their own channel. The multiplexer/demultiplexers need not be concerned with these matters.

Framing

So we have seen that a link control protocol is not needed to manage the overall TDM link. There is, however, a basic requirement for framing. Because we are not providing flag or SYNC characters to bracket TDM frames, some means is needed to assure frame synchronization. It is clearly important to maintain framing synchronization because, if the source and destination are out of step, data on all channels are lost.

Perhaps the most common mechanism for framing is known as added-digit framing. In this scheme, typically, one control bit is added to each TDM frame. An identifiable pattern of bits, from frame to frame, is used on this "control channel." A typical example is the alternating bit pattern, 101010 . . . . This is a pattern unlikely to be sustained on a data channel. Thus, to synchronize, a receiver compares the incoming bits of one frame position to the expected pattern. If the pattern does not match, successive bit positions are searched until the pattern persists over multiple frames. Once framing synchronization is established, the receiver continues to monitor the framing bit channel. If the pattern breaks down, the receiver must again enter a framing search mode.

Pulse Stuffing

Perhaps the most difficult problem in the design of a synchronous time-division multiplexer is that of synchronizing the various data sources. If each source has a separate clock, any variation among clocks could cause loss of synchronization. Also, in some cases, the data rates of the input data streams are not related by a simple rational number. For both these problems, a technique known as pulse stuffing is an effective remedy. With pulse stuffing, the outgoing data rate of the multiplexer, excluding framing bits, is higher than the sum of the maximum instantaneous incoming rates. The extra capacity is used by stuffing extra dummy bits or

pulses into each incoming signal until its rate is raised to that of a locally-generated clock signal. The stuffed pulses are inserted at fixed locations in the multiplexer frame format so that they may be identified and removed at the demultiplexer.

Example

An example, from [COUC95], illustrates the use of synchronous TDM to multiplex digital and analog sources. Consider that there are 11 sources to be multiplexed on a single link:

- Source 1: Analog, 2-kHz bandwidth.
- Source 2: Analog, 4-kHz bandwidth.
- Source 3: Analog, 2-kHz bandwidth.
- Sources 4–11: Digital, 7200 bps synchronous.

As a first step, the analog sources are converted to digital using PCM. Recall from Chapter 5 that PCM is based on the sampling theorem, which dictates that a signal be sampled at a rate equal to twice its bandwidth. Thus, the required sampling rate is 4000 samples per second for sources 1 and 3, and 8000 samples per second for source 2. These samples, which are analog (PAM), must then be quantized or digitized. Let us assume that 4 bits are used for each analog sample. For convenience, these three sources will be multiplexed first, as a unit. At a scan rate of 4 kHz, one PAM sample each is taken from sources 1 and 3, and two PAM samples are taken from source 2 per scan. These four samples are interleaved and converted to 4-bit PCM samples. Thus, a total of 16 bits is generated at a rate of 4000 times per second, for a composite bit rate of 64 kbps.

For the digital sources, pulse stuffing is used to raise each source to a rate of 8 kbps, for an aggregate data rate of 64 kbps. A frame can consist of multiple cycles of 32 bits, each containing 16 PCM bits and two bits from each of the eight digital sources. Figure 7.8 depicts the result.

Digital Carrier Systems

The long-distance carrier system provided in the United States and throughout the world was designed to transmit voice signals over high-capacity transmission links, such as optical fiber, coaxial cable, and microwave. Part of the evolution of these telecommunications networks toward digital technology has been the adoption of synchronous TDM transmission structures. In the United States, AT&T developed a hierarchy of TDM structures of various capacities; this structure is used in Canada and Japan as well as in the United States. A similar, but unfortunately not identical, hierarchy has been adopted internationally under the auspices of ITU-T (Table 7.3).

The basis of the TDM hierarchy (in North America and Japan) is the DS-1 transmission format (Figure 7.9), which multiplexes 24 channels. Each frame contains 8 bits per channel plus a framing bit for $24 \times 8 + 1 = 193$ bits. For voice transmission, the following rules apply. Each channel contains one word of digitized voice data. The original analog voice signal is digitized using pulse code modulation

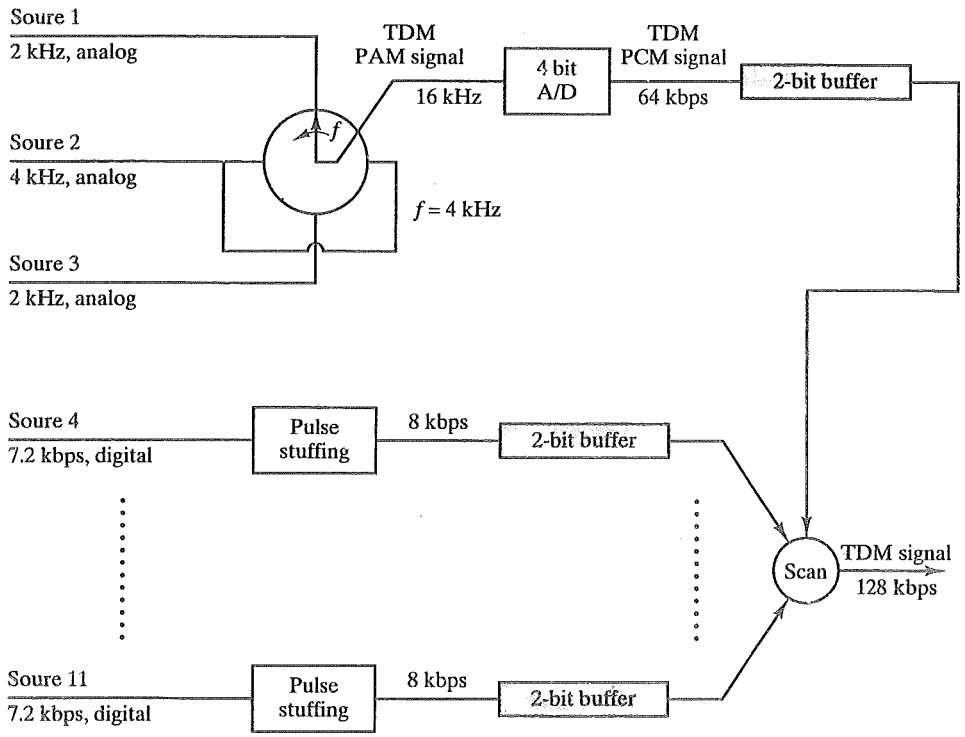


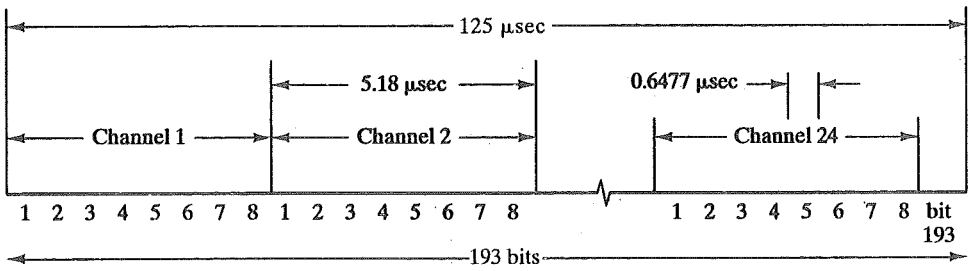
FIGURE 7.8 TDM of analog and digital sources.

(PCM) at a rate of 8000 samples per second. Therefore, each channel slot and, hence, each frame must repeat 8000 times per second. With a frame length of 193 bits, we have a data rate of 8000 × 193 = 1.544 Mbps. For five of every six frames, 8-bit PCM samples are used. For every sixth frame, each channel contains a 7-bit PCM word plus a *signaling bit*. The signaling bits form a stream for each voice channel that contains network control and routing information. For example, control signals are used to establish a connection or to terminate a call.

The same DS-1 format is used to provide digital data service. For compatibility with voice, the same 1.544-Mbps data rate is used. In this case, 23 channels of

TABLE 7.3 North American and international TDM carrier standards.

(a) North American			(b) International (ITU-T)		
Digital signal number	Number of voice channels	Data rate (Mbps)	Level number	Number of voice channels	Data rate (Mbps)
DS-1	24	1.544	1	30	2.048
DS-1C	48	3.152	2	120	8.448
DS-2	96	6.312	3	480	34.368
DS-3	672	44.736	4	1920	139.264
DS-4	4032	274.176	5	7680	565.148



Notes:

1. Bit 193 is a framing bit, used for synchronization.
2. Voice channels:
   - 8-bit PCM used on five of six frames.
   - 7-bit PCM used on every sixth frame. Bit 8 of each channel is a signaling bit.
3. Data channels:
   - Channel 24 used for signaling only in some schemes.
   - Bit 8 is a control bit.
   - Bits 1–7 used for 56 kbps service.
   - Bits 2–7 used for 9.6 kbps, 4.8 kbps, and 2.4 kbps service.

FIGURE 7.9 DS-1 transmission format.

data are provided. The twenty-fourth channel position is reserved for a special sync byte, which allows faster and more reliable reframing following a framing error. Within each channel, seven bits per frame are used for data, with the eighth bit used to indicate whether the channel, for that frame, contains user data or system control data. With seven bits per channel, and because each frame is repeated 8000 times per second, a data rate of 56 kbps can be provided per channel. Lower data rates are provided using a technique known as subrate multiplexing. For this technique, an additional bit is robbed from each channel to indicate which subrate multiplexing rate is being provided; this leaves a total capacity per channel of $6 \times 8000 = 48$ kbps. This capacity is used to multiplex five 9.6-kbps channels, ten 4.8-kbps channels, or twenty 2.4-kbps channels. For example, if channel 2 is used to provide 9.6-kbps service, then up to five data subchannels share this channel. The data for each subchannel appear as six bits in channel 2 every fifth frame.

Finally, the DS-1 format can be used to carry a mixture of voice and data channels. In this case, all 24 channels are utilized; no sync byte is provided.

Above this basic data rate of 1.544 Mbps, higher-level multiplexing is achieved by interleaving bits from DS-1 inputs. For example, the DS-2 transmission system combines four DS-1 inputs into a 6.312-Mbps stream. Data from the four sources are interleaved 12 bits at a time. Note that $1.544 \times 4 = 6.176$ Mbps. The remaining capacity is used for framing and control bits.

ISDN User-Network Interface

ISDN enables the user to multiplex traffic from a number of devices on the user's premises over a single line into an ISDN (Integrated Services Digital Network). Two interfaces are defined: a basic interface and a primary interface.

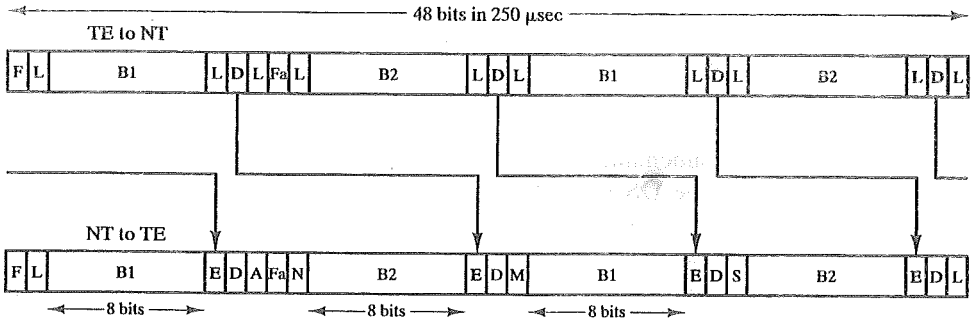
Basic ISDN Interface

At the interface between the subscriber and the network terminating equipment, digital data are exchanged using full-duplex transmission. A separate physical line is used for the transmission in each direction. The line coding specification for the interface dictates the use of a pseudoternary coding scheme.[1] Binary one is represented by the absence of voltage; binary zero is represented by a positive or negative pulse of 750 mV ±10%. The data rate is 192 kbps.

The basic access structure consists of two 64-kbps B channels and one 16-kbps D channel. These channels, which produce a load of 144 kbps, are multiplexed over a 192-kbps interface at the S or T reference point. The remaining capacity is used for various framing and synchronization purposes.

The B channel is the basic user channel. It can be used to carry digital data (e.g., a personal computer connection), PCM-encoded digital voice (e.g., a telephone connection), or any other traffic that can fit into a 64-kbps channel. At any given time, a logical connection can be set up separately for each B channel to separate ISDN destinations. The D channel can be used for a data-transmission connection at a lower data rate. It is also used to carry control information needed to set up and terminate the B-channel connections. Transmission on the D channel consists of a sequence of LAPD frames.

As with any synchronous time-division multiplexed (TDM) scheme, basic access transmission is structured into repetitive, fixed-length frames. In this case, each frame is 48 bits long; at 192 kbps, frames must repeat at a rate of one frame every 250 μsec. Figure 7.10 shows the frame structure; the upper frame is transmitted by the subscriber's terminal equipment (TE) to the network (NT); the lower frame is transmitted from the TE to the NT.



LEGEND

F = Framing bit
L = dc balancing bit
E = D-echo channel bit
A = Activation bit
Fa = Auxiliary framing bit
N = Set to opposite of Fa
M = Multiframing bit
B1= B channel bits (16 per frame)
B2 = B channel bits (16 per frame)
D = D channel bits (4 per frame)
S = Spare bits

FIGURE 7.10 Frame structure for ISDN basic rate access.

[1] See Section 4.1.

Each frame of 48 bits includes 16 bits from each of the two B channels and 4 bits from the D channel. The remaining bits have the following interpretation. Let us first consider the frame structure in the TE-to-NT direction. Each frame begins with a framing bit (F) that is always transmitted as a positive pulse. This is followed by a dc balancing bit (L) that is set to a negative pulse to balance the voltage. The F-L pattern thus acts to synchronize the receiver on the beginning of the frame. The specification dictates that, following these first two bit positions, the first occurrence of a zero bit will be encoded as a negative pulse. After that, the pseudoternary rules are observed. The next eight bits (B1) are from the first B channel; this is followed by another dc balancing bit (L). Next comes a bit from the D channel, followed by its balancing bit. This is followed by the auxiliary framing bit (F_A), which is set to zero unless it is to be used in a multiframe structure. There follows another balancing bit (L), eight bits (B2) from the second B channel, and another balancing bit (L); this is followed by bits from the D channel, first B channel, D channel again, second B channel, and the D channel yet again, with each group of channel bits followed by a balancing bit.

The frame structure in the NT-to-TE direction is similar to the frame structure for transmission in the TE-to-NT direction. The following new bits replace some of the dc balancing bits. The D-channel echo bit (E) is a retransmission by the NT of the most recently received D bit from the TE; the purpose of this echo is explained below. The activation bit (A) is used to activate or deactivate a TE, allowing the device to come on line or, when there is no activity, to be placed in low-power-consumption mode. The N bit is normally set to binary one. The N and M bits may be used for multiframing. The S bit is reserved for other future standardization requirements.

The E bit in the TE-to-NT direction comes into play to support a contention resolution function, which is required when multiple TE1 terminals share a single physical line (i.e., a multipoint line). There are three types of traffic to consider:

- **B-channel traffic.** No additional functionality is needed to control access to the two B channels, as each channel is dedicated to a particular TE at any given time.
- **D-channel traffic.** The D channel is available for use by all the subscriber devices for both control signaling and packet transmission, so the potential for contention exists. There are two subcases:
  - *Incoming traffic:* The LAPD addressing scheme is sufficient to sort out the proper destination for each data unit.
  - *Outgoing traffic:* Access must be regulated so that only one device at a time transmits. This is the purpose of the contention-resolution algorithm.

The D-channel contention-resolution algorithm has the following elements:

1. When a subscriber device has no LAPD frames to transmit, it transmits a series of binary ones on the D channel; using the pseudoternary encoding scheme, this corresponds to the absence of line signal.
2. The NT, on receipt of a D-channel bit, reflects back the binary value as a D-channel echo bit.

3. When a terminal is ready to transmit an LAPD frame, it listens to the stream of incoming D-channel echo bits. If it detects a string of 1-bits equal in length to a threshold value X_i, it may transmit; otherwise, the terminal must assume that some other terminal is transmitting, and wait.
4. It may happen that several terminals are monitoring the echo stream and begin to transmit at the same time, causing a collision. To overcome this condition, a transmitting TE monitors the E bits and compares them to its transmitted D bits. If a discrepancy is detected, the terminal ceases to transmit and returns to a listen state.

The electrical characteristics of the interface (i.e., 1-bit = absence of signal) are such that any user equipment transmitting a 0-bit will override user equipment transmitting a 1-bit at the same instant. This arrangement ensures that one device will be guaranteed successful completion of its transmission.

The algorithm includes a primitive priority mechanism based on the threshold value X_i. Control information is given priority over user data. Within each of these two priority classes, a station begins at normal priority and then is reduced to lower priority after a transmission. It remains at the lower priority until all other terminals have had an opportunity to transmit. The values of X_i are as follows:

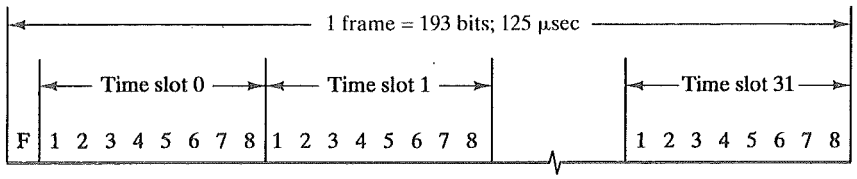
- Control Information
  - Normal priority $X_1 = 8$
  - Lower priority $X_1 = 9$
- User Data
  - Normal priority $X_2 = 10$
  - Lower priority $X_2 = 11$

Primary ISDN Interface

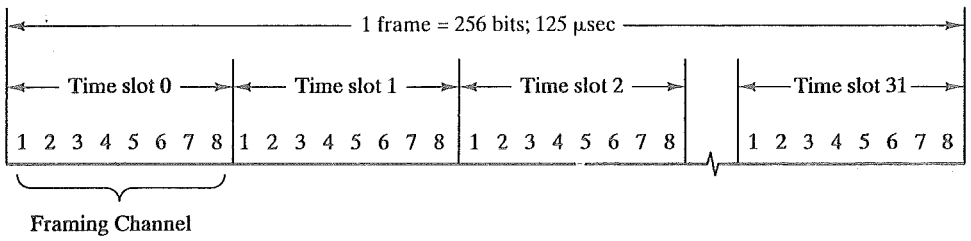
The primary interface, like the basic interface, multiplexes multiple channels across a single transmission medium. In the case of the primary interface, only a point-to-point configuration is allowed. Typically, the interface supports a digital PBX or other concentration device controlling multiple TEs and providing a synchronous TDM facility for access to ISDN. Two data rates are defined for the primary interface: 1.544 Mbps and 2.048 Mbps.

The ISDN interface at 1.544 Mbps is based on the North American DS-1 transmission structure, which is used on the T1 transmission service. Figure 7.11a illustrates the frame format for this data rate. The bit stream is structured into repetitive 193-bit frames. Each frame consists of 24 8-bit time slots and a framing bit, which is used for synchronization and other management purposes. The same time slot repeated over multiple frames constitutes a channel. At a data rate of 1.544 Mbps, frames repeat at a rate of one every 125 μsec, or 8000 frames per second. Thus, each channel supports 64 kbps. Typically, the transmission structure is used to support 23 B channels and 1 64-kbps D channel.

The line coding for the 1.544-Mbps interface is AMI (Alternate Mark Inversion) using B8ZS.



(a) Interface at 1.544 Mbps



(b) Interface at 2.048 Mbps

FIGURE 7.11 ISDN primary access frame formats.

The ISDN interface at 2.048 Mbps is based on the European transmission structure of the same data rate. Figure 7.11b illustrates the frame format for this data rate. The bit stream is structured into repetitive 256-bit frames. Each frame consists of 32 8-bit time slots. The first time slot is used for framing and synchronization purposes; the remaining 31 time slots support user channels. At a data rate of 2.048 Mbps, frames repeat at a rate of one every 125 μsec, or 800 frames per second. Thus, each channel supports 64 kbps. Typically, the transmission structure is used to support 30 B channels and 1 D channel.

The line coding for the 2.048-Mbps interface is AMI using HDB3.

SONET/SDH

SONET (Synchronous Optical Network) is an optical transmission interface originally proposed by BellCore and standardized by ANSI. A compatible version, referred to as Synchronous Digital Hierarchy (SDH), has been published by ITU-T in Recommendations G.707, G.708, and G.709.[2] SONET is intended to provide a specification for taking advantage of the high-speed digital transmission capability of optical fiber.

Signal Hierarchy

The SONET specification defines a hierarchy of standardized digital data rates (Table 7.4). The lowest level, referred to as STS-1 (Synchronous Transport Signal,

[2] In what follows, we will use the term SONET to refer to both specifications. Differences that exist will be addressed.

TABLE 7.4 SONET/SDH signal hierarchy.

SONET designation	CCITT designation	Data rate (MBPS)	Payload rate (Mbps)
STS-1/OC-1		51.84	50.112
STS-3/OC-3	STM-1	155.52	150.336
STS-9/OC-9	STM-3	466.56	451.008
STS-12/OC-12	STM-4	622.08	601.344
STS-18/OC-18	STM-6	933.12	902.016
STS-24/OC-24	STM-8	1244.16	1202.688
STS-36/OC-36	STM-12	1866.24	1804.032
STS-48/OC-48	STM-16	2488.32	2405.376

level 1) or OC-1 (Optical Carrier level 1),[3] is 51.84 Mbps. This rate can be used to carry a single DS-3 signal or a group of lower-rate signals, such as DS1, DS1C, DS2, plus ITU-T rates (e.g., 2.048 Mbps).

Multiple STS-1 signals can be combined to form an STS-N signal. The signal is created by interleaving bytes from N STS-1 signals that are mutually synchronized.

For the ITU-T Synchronous Digital Hierarchy, the lowest rate is 155.52 Mbps, which is designated STM-1. This corresponds to SONET STS-3. The reason for the discrepancy is that STM-1 is the lowest-rate signal that can accommodate an ITU-T level 4 signal (139.264 Mbps).

Frame Format

The basic SONET building block is the STS-1 frame, which consists of 810 octets and is transmitted once every 125 μs, for an overall data rate of 51.84 Mbps (Figure 7.12a). The frame can logically be viewed as a matrix of 9 rows of 90 octets each, with transmission being one row at a time, from left to right and top to bottom.

The first three columns (3 octets $\times$ 9 rows = 27 octets) of the frame are devoted to overhead octets. Nine octets are devoted to section-related overhead and 18 octets are devoted to line overhead. Figure 7.13a shows the arrangement of overhead octets, and Table 7.5 defines the various fields.

The remainder of the frame is payload, which is provided by the path layer. The payload includes a column of path overhead, which is not necessarily in the first available column position; the line overhead contains a pointer that indicates where the path overhead starts. Figure 7.13b shows the arrangement of path overhead octets, and Table 7.5 defines these.

Figure 7.12b shows the general format for higher-rate frames, using the ITU-T designation.

[3] An OC-N rate is the optical equivalent of an STS-N electrical signal. End user devices transmit and receive electrical signals; these must be converted to and from optical signals for transmission over optical fiber.

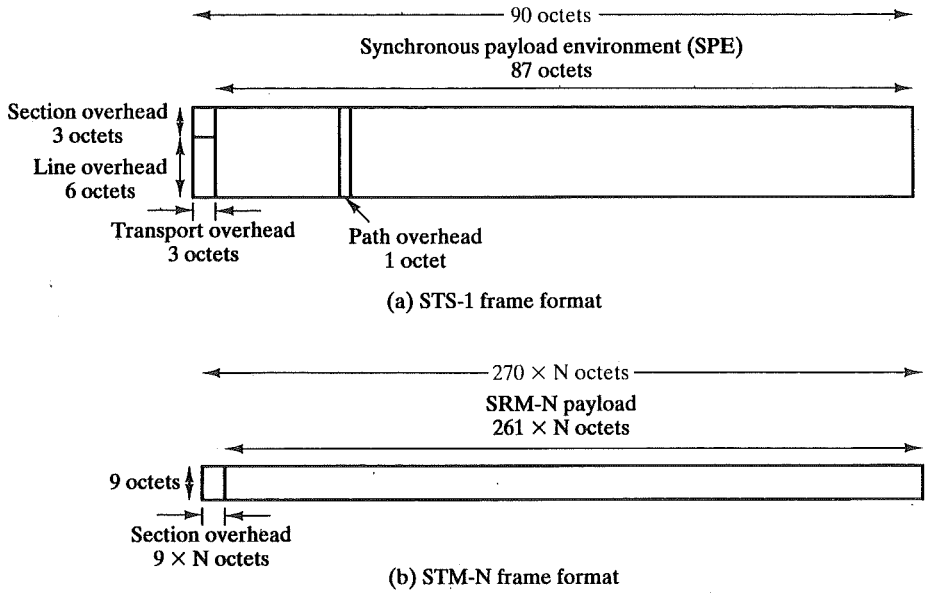


FIGURE 7.12 SONET/SDH frame formats.

Section Overhead	Framing A1	Framing A2	STS-ID C1
	BIP-8 B1	Orderwire E1	User F1
	Data Com D1	Data Com D2	Data Com D3
Line Overhead	Pointer H1	Pointer H2	Pointer Action H3
	BIP-8 B2	APS K1	APS K2
	Data Com D4	Data Com D5	Data Com D6
	Data Com D7	Data Com D8	Data Com D9
	Data Com D10	Data Com D11	Data Com D12
	Growth Z1	Growth Z2	Orderwire E2

(a) Section overhead

Trace J1
BIP-8 B3
Signal Label C2
Path Status G1
User F2
Multiframe H4
Growth Z3
Growth Z4
Growth Z5

(b) Path overhead

FIGURE 7.13 SONET STS-1 overhead octets.

TABLE 7.5 STS-1 Overhead bits.

	Section overhead
A1, A2:	Framing bytes = F6,28 hex; used to synchronize the beginning of the frame.
C1:	STS-1 ID identifies the STS-1 number (1 to N) for each STS-1 within an STS-*N* multiplex.
B1:	Bit-interleaved parity byte providing even parity over previous STS-*N* frame after scrambling; the *i*th bit of this octet contains the even parity value calculated from the ith bit position of all octets in the previous frame.
E1:	Section level 64-kbps PCM orderwire; optional 64 Kbps voice channel to be used between section terminating equipment, hubs, and remote terminals.
F1:	64-kbps channel set aside for user purposes.
D1–D3:	192-kbps data communications channel for alarms, maintenance, control, and administration between sections.
	Line overhead
H1–H3:	Pointer bytes used in frame alignment and frequency adjustment of payload data.
B2:	Bit-interleaved parity for line level error monitoring.
K1, K2:	Two byes allocated for signaling between line level automatic protection switching equipment; uses a bit-oriented protocol that provides for error protection and management of the SONET optical link.
D4–D12:	576-kbps data communications channel for alarms, maintenance, control, monitoring, and administration at the line level.
Z1, Z2	Reserved for future use.
E2:	64-kbps PCM voice channel for line level orderwire.
	Path overhead
J1:	64-kbps channel used to repetitively send a 64-octet fixed-length string so a receiving terminal can continuously verify the integrity of a path; the contents of the message are user programmable.
B3:	Bit-interleaved parity at the path level, calculated over all bits of the previous SPE.
C2:	STS path signal label to designate equipped versus unequipped STS signals. Unequipped means the the line connection is complete but there is no path data to send. For equipped signals, the label can indicate the specific STS payload mapping that might be needed in receiving terminals to interpret the payloads.
G1:	Status byte sent from path terminating equipment back to path originating equipment to convey status of terminating equipment and path error performance.
F2:	64-kbps channel for path user.
H4:	Multiframe indicator for payloads needing frames that are longer than a single STS frame; multiframe indicators are used when packing lower rate channels (virtual tributaries) into the SPE.
Z3–Z5:	Reserved for future use.

7.3 STATISTICAL TIME-DIVISION MULTIPLEXING

Characteristics

In a synchronous time-division multiplexer, it is generally the case that many of the time slots in a frame are wasted. A typical application of a synchronous TDM involves linking a number of terminals to a shared computer port. Even if all terminals are actively in use, most of the time there is no data transfer at any particular terminal.

An alternative to synchronous TDM is statistical TDM, also known as asynchronous TDM and intelligent TDM. The statistical multiplexer exploits this common property of data transmission by dynamically allocating time slots on demand. As with a synchronous TDM, the statistical multiplexer has a number of I/O lines on one side and a higher-speed multiplexed line on the other. Each I/O line has a buffer associated with it. In the case of the statistical multiplexer, there are n I/O lines, but only k, where $k < n$, time slots available on the TDM frame. For input, the function of the multiplexer is to scan the input buffers, collecting data until a frame is filled, and then send the frame. On output, the multiplexer receives a frame and distributes the slots of data to the appropriate output buffers.

Because statistical TDM takes advantage of the fact that the attached devices are not all transmitting all of the time, the data rate on the multiplexed line is less than the sum of the data rates of the attached devices. Thus, a statistical multiplexer can use a lower data rate to support as many devices as a synchronous multiplexer. Alternatively, if a statistical multiplexer and a synchronous multiplexer both use a link of the same data rate, the statistical multiplexer can support more devices.

Figure 7.14 contrasts statistical and synchronous TDM. The figure depicts four data sources and shows the data produced in four time epochs (t_0, t_1, t_2, t_3). In the case of the synchronous multiplexer, the multiplexer has an effective output rate of four times the data rate of any of the input devices. During each epoch, data are collected from all four sources and sent out. For example, in the first epoch, sources C and D produce no data. Thus, two of the four time slots transmitted by the multiplexer are empty.

In contrast, the statistical multiplexer does not send empty slots if there are data to send. Thus, during the first epoch, only slots for A and B are sent. However, the positional significance of the slots is lost in this scheme. It is not known ahead of time which source's data will be in any particular slot. Because data arrive from

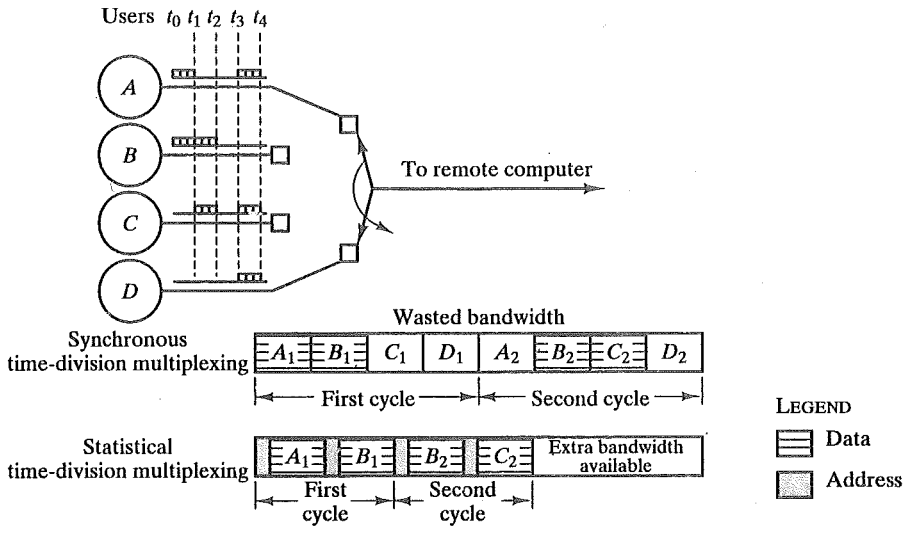


FIGURE 7.14 Synchronous TDM contrasted with statistical TDM.

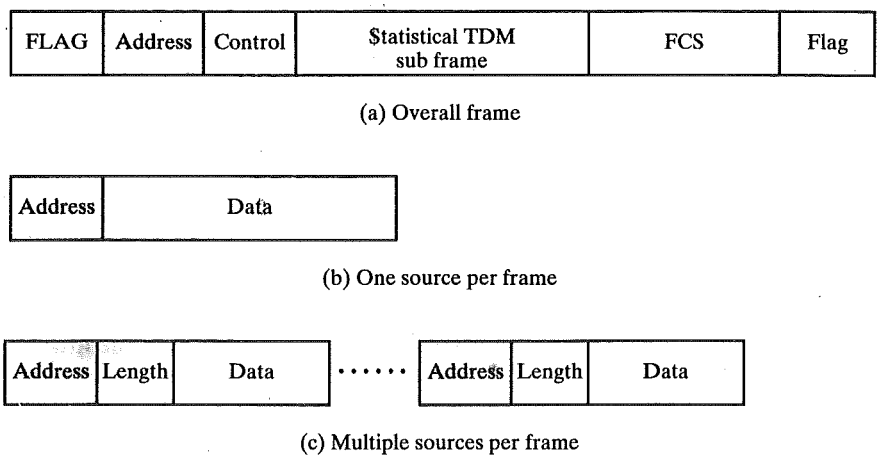


FIGURE 7.15 Statistical TDM frame formats.

and are distributed to I/O lines unpredictably, address information is required to assure proper delivery. As a result, there is more overhead per slot for statistical TDM as each slot carries an address as well as data.

The frame structure used by a statistical multiplexer has an impact on performance. Clearly, it is desirable to minimize overhead bits to improve throughput. Generally, a statistical TDM system will use a synchronous protocol such as HDLC. Within the HDLC frame, the data frame must contain control bits for the multiplexing operation. Figure 7.15 shows two possible formats. In the first case, only one source of data is included per frame. That source is identified by an address. The length of the data field is variable, and its end is marked by the end of the overall frame. This scheme can work well under light load, but is quite inefficient under heavy load.

A way to improve efficiency is to allow multiple data sources to be packaged in a single frame. Now, however, some means is needed to specify the length of data for each source. Thus, the statistical TDM subframe consists of a sequence of data fields, each labeled with an address and a length. Several techniques can be used to make this approach even more efficient. The address field can be reduced by using relative addressing. That is, each address specifies the number of the current source relative to the previous source, modulo the total number of sources. So, for example, instead of an 8-bit address field, a 4-bit field might suffice.

Another refinement is to use a two-bit label with the length field. A value of 00, 01, or 10 corresponds to a data field of one, two, or three bytes; no length field is necessary. A value of 11 indicates that a length field is included.

Performance

We have said that the data rate of the output of a statistical multiplexer is less than the sum of the data rates of the inputs. This is allowable because it is anticipated that the average amount of input is less than the capacity of the multiplexed line.

The difficulty with this approach is that, while the average aggregate input may be less than the multiplexed line capacity, there may be peak periods when the input exceeds capacity.

The solution to this problem is to include a buffer in the multiplexer to hold temporary excess input. Table 7.6 gives an example of the behavior of such systems. We assume 10 sources, each capable of 1000 bps, and we assume that the average input per source is 50% of its maximum. Thus, on average, the input load is 5000 bps. Two cases are shown: multiplexers of output capacity 5000 bps and 7000 bps. The entries in the table show the number of bits input from the 10 devices each millisecond and the output from the multiplexer. When the input exceeds the output, backlog develops that must be buffered.

There is a trade-off between the size of the buffer used and the data rate of the line. We would like to use the smallest possible buffer and the smallest possible data rate, but a reduction in one requires an increase in the other. Note that we are not so much concerned with the cost of the buffer—memory is cheap—as we are with the fact that the more buffering there is, the longer the delay. Thus, the trade-off is really one between system response time and the speed of the multiplexed line. In this section, we present some approximate measures that examine this trade-off. These are sufficient for most purposes.

Let us define the following parameters for a statistical time-division multiplexer:

TABLE 7.6 Example of statistical multiplexer performance.

	Capacity = 5000 bps		Capacity = 7000 bps	
Input[a]	Output	Backlog	Output	Backlog
6	5	1	6	0
9	5	5	7	2
3	5	3	5	0
7	5	5	7	0
2	5	2	2	0
2	4	0	2	0
2	2	0	2	0
3	3	0	3	0
4	4	0	4	0
6	5	1	6	0
1	2	0	1	0
10	5	5	7	3
7	5	7	7	3
5	5	7	7	1
8	5	10	7	2
3	5	8	5	0
6	5	9	6	0
2	5	6	2	0
9	5	10	7	2
5	5	10	7	0

[a] Input = 10 sources, 1000 bps/source; average input rate = 50% of maximum.

N = number of input sources
R = data rate of each source, bps
M = effective capacity of multiplexed line, bps
α = mean fraction of time each source is transmitting, $0 < \alpha < 1$
$K = \frac{M}{NR}$ = ratio of multiplexed line capacity to total maximum input

In the above, we have defined M taking into account the overhead bits introduced by the multiplexer. That is, M represents the maximum rate at which data bits can be transmitted.

The parameter K is a measure of the compression achieved by the multiplexer. For example, for a given data rate M, if $K = 0.25$, there are four times as many devices being handled as by a synchronous time-division multiplexer using the same link capacity. The value of K can be bounded:

$$\alpha < K < 1$$

A value of $K = 1$ corresponds to a synchronous time-division multiplexer, as the system has the capacity to service all input devices at the same time. If $K < \alpha$, the input will exceed the multiplexer's capacity.

Some results can be obtained by viewing the multiplexer as a single-server queue. A queuing situation arises when a "customer" arrives at a service facility and, finding it busy, is forced to wait. The delay incurred by a customer is the time spent waiting in the queue plus the time for the service. The delay depends on the pattern of arriving traffic and the characteristics of the server. Table 7.7 summarizes results for the case of random (Poisson) arrivals and constant service time. This model is easily related to the statistical multiplexer:

TABLE 7.7 Single-server queues with constant service times and poisson (random) arrivals.

Parameters

λ = mean number of arrivals per second
s = service time for each arrival
ρ = utilization, fraction of time the server is busy
q = mean number of items in system (waiting and being served)
t_q = mean time an item spends in system
σq = standard deviation of q

Formulas

$$\rho = \lambda s$$

$$q = \frac{\rho^2}{2(1-\rho)} + \rho$$

$$t_q = \frac{s(2-\rho)}{2(1-\rho)}$$

$$\sigma q = \frac{1}{1-p}\sqrt{\rho - \frac{3\rho^2}{2} + \frac{5\rho^3}{6} - \frac{\rho^4}{12}}$$

$$\lambda = \alpha NR$$
$$S = \frac{1}{M}$$

The average arrival rate λ, in bps, is the total potential input (NR) times the fraction of time α that each source is transmitting. The service time S, in seconds, is the time it takes to transmit one bit, which is $1/M$. Note that

$$\rho = \lambda S = \frac{\alpha NR}{M} = \frac{\alpha}{K} = \frac{\lambda}{M}$$

The parameter ρ is the utilization or fraction of total link capacity being used. For example, if the capacity M is 50 kbps and $\rho = 0.5$, the load on the system is 25 kbps. The parameter q is a measure of the amount of buffer space being used in the multiplexer. Finally, t_q is a measure of the average delay encountered by an input source.

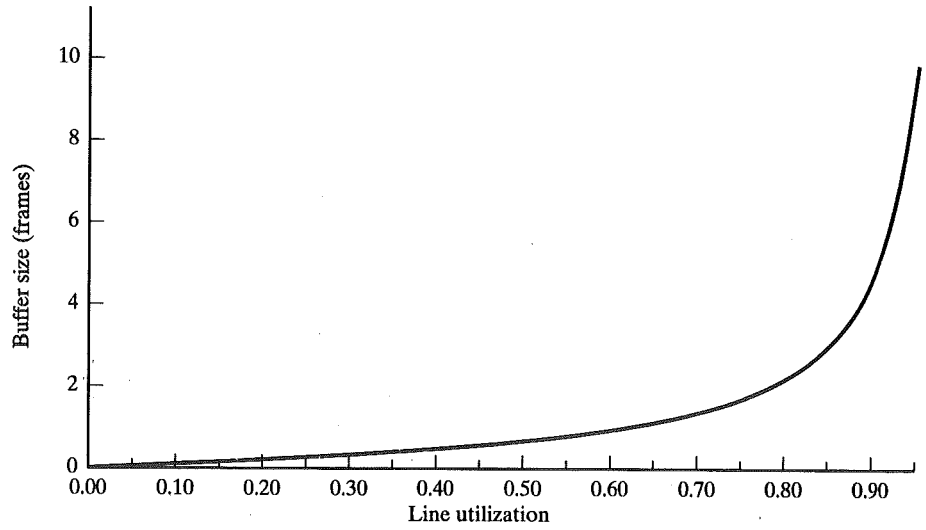
Figure 7.16 gives some insight into the nature of the trade-off between system response time and the speed of the multiplexed line. It assumes that data are being transmitted in 1000-bit frames. Part (a) of the figure shows the average number of frames that must be buffered as a function of the average utilization of the multiplexed line. The utilization is expressed as a percentage of the total line capacity. Thus, if the average input load is 5000 bps, the utilization is 100 percent for a line capacity of 5000 bps and about 71 percent for a line capacity of 7000 bps. Part (b) of the figure shows the average delay experienced by a frame as a function of utilization and data rate. Note that as the utilization rises, so do the buffer requirements and the delay. A utilization above 80 percent is clearly undesirable.

Note that the average buffer size being used depends only on ρ, and not directly on M. For example, consider the following two cases:

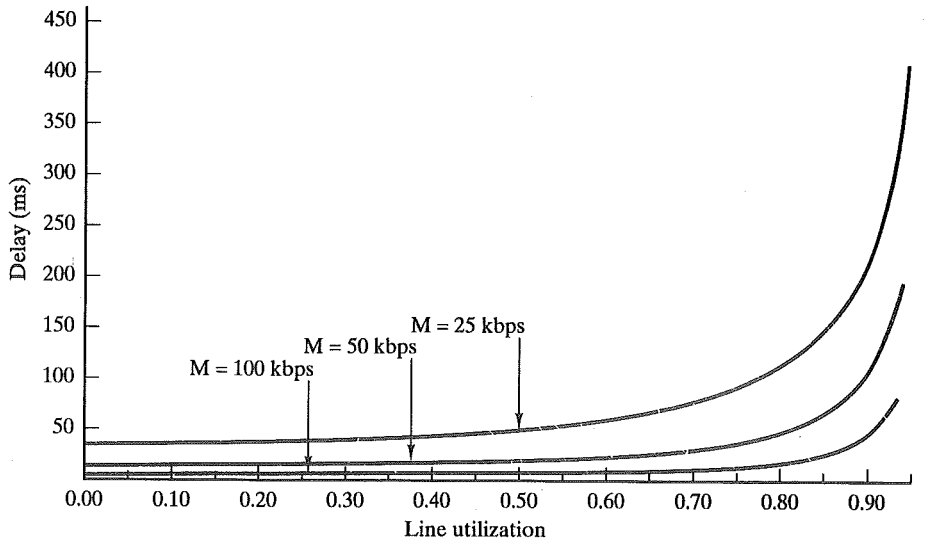
Case I	Case II
$N = 10$	$N = 100$
$R = 100$bps	$R = 100$ bps
$\alpha = 0.4$	$\alpha = 0.4$
$M = 500$ bps	$M = 5000$ bps

In both cases, the value of ρ is 0.8 and the mean buffer size is 2.4. Thus, proportionately, a smaller amount of buffer space per source is needed for multiplexers that handle a larger number of sources. Figure 7.16b also shows that the average delay will be smaller as the link capacity increases, for constant utilization.

So far, we have been considering average queue length, and, hence, the average amount of buffer capacity needed. Of course, there will be some fixed upper bound on the buffer size available. The variance of the queue size grows with utilization. Thus, at a higher level of utilization, a larger buffer is needed to hold the



(a) Mean buffer size versus utilization



(b) Mean delay versus utilization

FIGURE 7.16 Buffer size and delay for a statistical multiplexer.

backlog. Even so, there is always a finite probability that the buffer will overflow. Figure 7.17 shows the strong dependence of overflow probability on utilization. This figure, plus Figure 7.16, suggest that utilization above about 0.8 is undesirable.

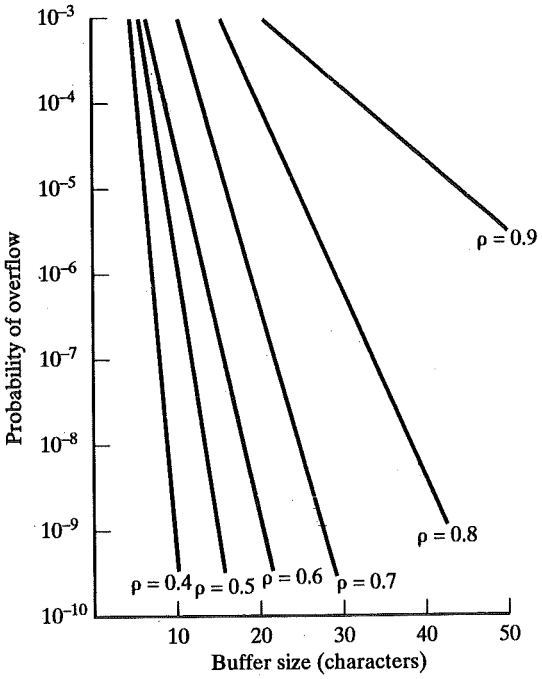


FIGURE 7.17 Probability of overflow as a function of buffer size.

7.4 RECOMMENDED READING

A discussion of FDM and TDM carrier systems can be found in [BELL90] and [FREE94]. More detailed description and analysis of TDM carrier systems is provided by [POWE90]. ISDN interfaces and SONET are treated in greater depth in [STAL95].

BELL90 Bellcore (Bell Communications Research). *Telecommunications Transmission Engineering*. Three volumes. 1990.

FREE94 Freeman, R. *Reference Manual for Telecommunications Engineering*. New York: Wiley, 1994.

POWE90 Powers, J. and Stair, H. *Megabit Data Communications*. Englewood Cliffs, NJ: Prentice Hall, 1990.

STAL95 Stallings, W. *ISDN and Broadband ISDN, with Frame Relay and ATM*. Englewood Cliffs, NJ: Prentice Hall, 1995.

Recommended Web Site

- http://www.atis.org/sif/sifhom.html: SONET Interoperability Forum site. Discusses current projects and technology.

7.5 PROBLEMS

7.1 The information in four analog signals is to be multiplexed and transmitted over a telephone channel that has a 400- to 3100-Hz bandpass. Each of the analog baseband signals is bandlimited to 500 Hz. Design a communication system (block diagram) that will allow the transmission of these four sources over the telephone channel using
a. Frequency-division multiplexing with SSB (single sideband) subcarriers.
b. Time-division multiplexing using PCM.

Show the block diagrams of the complete system, including the transmission, channel, and reception portions. Include the bandwidths of the signals at the various points in the systems.

7.2 To paraphrase Lincoln, All of the channel some of the time, some of the channel all of the time. Refer to Figure 7.2 and relate the preceding to the figure.

7.3 Consider a transmission system using frequency-division multiplexing. What cost factors are involved in adding one more pair of stations to the system?

7.4 Ten analog signals that are bandlimited to frequencies below 16 kHz are sampled at the Nyquist rate. The digitizing error is to be held below 0.2%. The signals are to travel on a synchronous TDM channel. What is the data rate required for the channel?

7.5 In Synchronous TDM, it is possible to interleave bits, one bit from each channel participating in a cycle. If the channel is using a self-clocking code in order to assist synchronization, might this bit interleaving introduce problems, as there is not a continuous stream of bits from one source?

7.6 Why is it that the start and stop bits can be eliminated when character interleaving is used in synchronous TDM?

7.7 Explain in terms of data link control and physical-layer concepts how error and flow control are accomplished in synchronous time-division multiplexing.

7.8 Bit 193 in the DS-1 transmission format is used for frame synchronization. Explain its use.

7.9 In the DS-1 format, what is the control signal data rate for each voice channel?

7.10 Twenty-four voice signals are to be multiplexed and transmitted over twisted pair. What is the bandwidth required for FDM? Assuming a bandwidth efficiency of 1 bps/Hz, what is the bandwidth required for TDM using PCM?

7.11 Draw a block diagram similar to Figure 7.8 for a TDM PCM system that will accommodate four 300-bps, synchronous, digital inputs and one analog input with a bandwidth of 500 Hz. Assume that the analog samples will be encoded into 4-bit PCM words.

7.12 A character-interleaved time-division multiplexer is used to combine the data streams of a number of 110-bps asynchronous terminals for data transmission over a 2400-bps digital line. Each terminal sends characters consisting of 7 data bits, 1 parity bit, 1 start bit, and 2 stop bits. Assume that one synchronization character is sent every 19 data characters and, in addition, at least 3% of the line capacity is reserved for pulse stuffing to accommodate speed variations from the various terminals.
a. Determine the number of bits per character.
b. Determine the number of terminals that can be accommodated by the multiplexer.
c. Sketch a possible framing pattern for the multiplexer.

7.13 Assume that two 600-bps terminals, five 300-bps terminals, and a number of 150-bps terminals are to be time-multiplexed in a character-interleaved format over a 4800-bps digital line. The terminals send 10 bits/character, and one synchronization character is inserted for every 99 data characters. All the terminals are asynchronous, and 3% of the line capacity is allocated for pulse stuffing to accommodate variations in the terminal clock rates.

a. Determine the number of 150-bps terminals that can be accommodated.
b. Sketch a possible framing pattern for the multiplexer.

7.14 Find the number of the following devices that could be accommodated by a T1-type TDM line if 1% of the line capacity is reserved for synchronization purposes.
a. 110-bps teleprinter terminals.
b. 300-bps computer terminals.
c. 1200-bps computer terminals.
d. 9600-bps computer output ports.
e. 64-kbps PCM voice-frequency lines.

How would these numbers change if each of the sources were operational an average of 10% of the time?

7.15 Ten 9600-bps lines are to be multiplexed using TDM. Ignoring overhead bits, what is the total capacity required for synchronous TDM? Assuming that we wish to limit average line utilization of 0.8, and assuming that each line is busy 50% of the time, what is the capacity required for statistical TDM?

7.16 For a statistical time-division multiplexer, define the following parameters:

F = frame length, bits
OH = overhead in a frame, bits
L = load of data in the frame, bps
C = capacity of link, bps

a. Express F as a function of the other parameters. Explain why F can be viewed as a variable rather than a constant.
b. Plot F versus L for $C = 9.6$ kbps and values of $OH = 40, 80, 120$. Comment on the results and compare to Figure 7.16.
c. Plot F versus L for $OH = 40$ and values of $C = 9.6$ kbps and 7.2 kbps. Comment on the results and compare to Figure 7.16.

7.17 The Clambake Zipper Company has two locations. The international headquarters is located at Cut and Shoot, Texas, while the factory is at Conroe, about 25 miles away. The factory has four 300-bps terminals that communicate with the central computer facilities at headquarters over leased voice grade lines. The company is considering installing time-division multiplexing equipment so that only one line will be needed. What cost factors should be considered in the decision?

7.18 In statistical TDM, there may be a length field. What alternative could there be to the inclusion of a length field? What problem might this solution cause and how could it be solved?

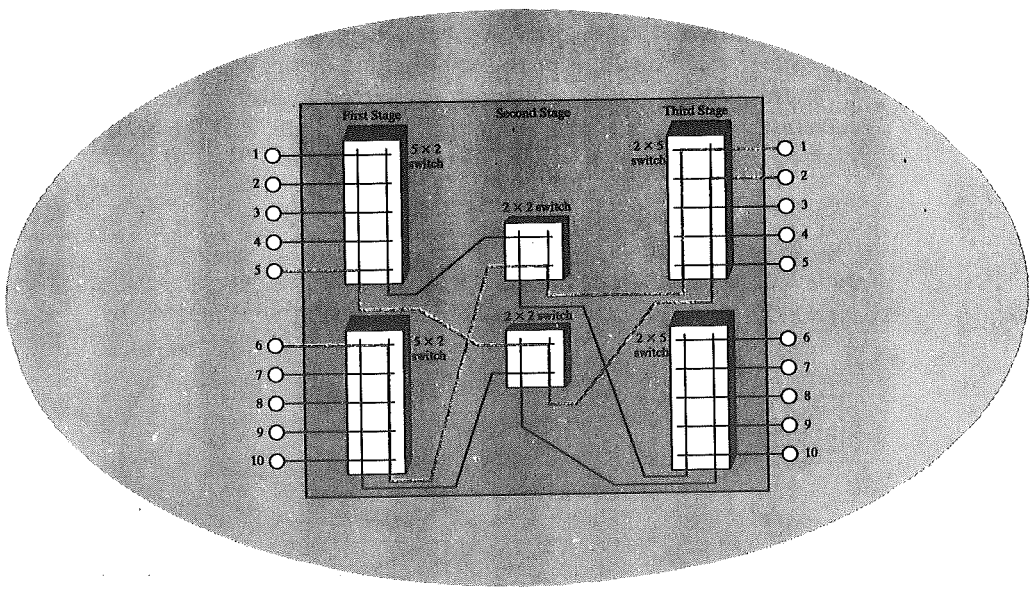
7.19 In synchronous TDM, the I/O lines serviced by the two multiplexers may be either synchronous or asynchronous, although the channel between the two multiplexers must be synchronous. Is there any inconsistency in this? Why or why not?

7.20 Assume that you are to design a TDM carrier—say, DS-489—to support 30 voice channels using 6 bit samples and a structure similar to DS-1. Determine the required bit rate.

PART TWO Wide-Area Networks

CHAPTER 8

CIRCUIT SWITCHING



Since the invention of the telephone, circuit switching has been the dominant technology for voice communications, and it will remain so well into the ISDN era. This chapter begins with an introduction to the concept of a switched communications network and then looks at the key characteristics of a circuit-switching network.

8.1 SWITCHING NETWORKS

For transmission of data[1] beyond a local area, communication is typically achieved by transmitting data from source to destination through a network of intermediate switching nodes; this switched-network design is sometimes used to implement LANs and MANs as well. The switching nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. Figure 8.1 illustrates a simple network. The end devices that wish to communicate may be referred to as *stations*. The stations may be computers, terminals, telephones, or other communicating devices. We will refer to the switching devices whose purpose is to provide communication as *nodes*, which are connected to each other in some topology by transmission links. Each station attaches to a node, and the collection of nodes is referred to as a *communications network*.

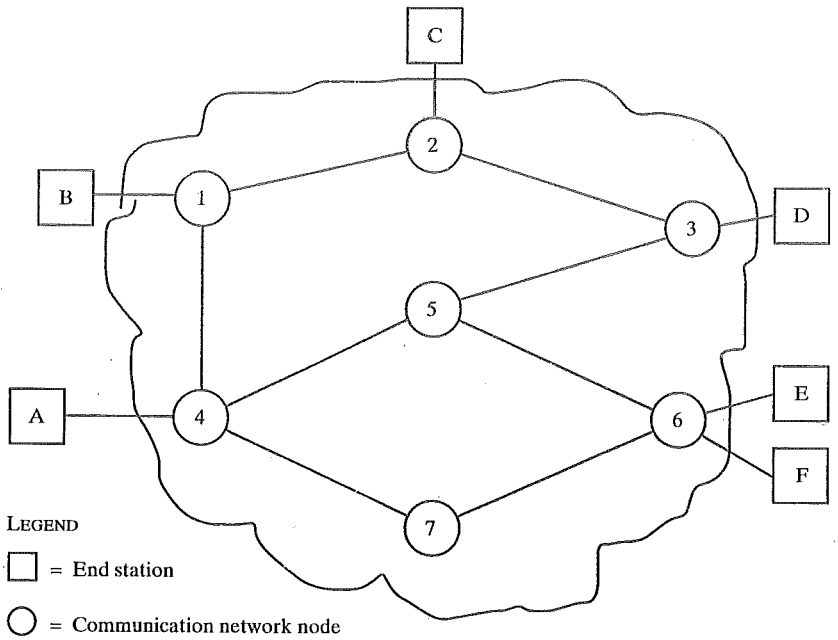


FIGURE 8.1 Simple switching network.

[1] We use this term here in a very general sense to include voice, image, and video, as well as ordinary data (e.g., numerical, text).

The types of networks that are discussed in this and the next three chapters are referred to as *switched communication networks*. Data entering the network from a station are routed to the destination by being switched from node to node. For example, in Figure 8.1, data from station A intended for station F are sent to node 4. They may then be routed via nodes 5 and 6 or nodes 7 and 6 to the destination. Several observations are in order:

1. Some nodes connect only to other nodes (e.g., 5 and 7). Their sole task is the internal (to the network) switching of data. Other nodes have one or more stations attached as well; in addition to their switching functions, such nodes accept data from and deliver data to the attached stations.
2. Node-node links are usually multiplexed, using either frequency-division multiplexing (FDM) or time-division multiplexing (TDM).
3. Usually, the network is not fully connected; that is, there is not a direct link between every possible pair of nodes. However, it is always desirable to have more than one possible path through the network for each pair of stations; this enhances the reliability of the network.

Two quite different technologies are used in wide-area switched networks: circuit switching and packet switching. These two technologies differ in the way the nodes switch information from one link to another on the way from source to destination. In this chapter, we look at the details of circuit switching; packet switching is pursued in Chapter 9. Two approaches that evolved from packet switching, namely frame relay and ATM, are explored in Chapters 10 and 11, respectively.

8.2 CIRCUIT-SWITCHING NETWORKS

Communication via circuit switching implies that there is a dedicated communication path between two stations. That path is a connected sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Communication via circuit switching involves three phases, which can be explained with reference to Figure 8.1.

1. *Circuit establishment.* Before any signals can be transmitted, an end-to-end (station-to-station) circuit must be established. For example, station *A* sends a request to node 4 requesting a connection to station *E*. Typically, the link from *A* to 4 is a dedicated line, so that part of the connection already exists. Node 4 must find the next leg in a route leading to node 6. Based on routing information and measures of availability and, perhaps, cost, node 4 selects the link to node 5, allocates a free channel (using frequency-division multiplexing, FDM, or time-division multiplexing, TDM) on that link and sends a message requesting connection to *E*. So far, a dedicated path has been established from A through 4 to 5. Because a number of stations may attach to 4, it must be able to establish internal paths from multiple stations to multiple nodes. The remainder of the process proceeds similarly. Node 5 dedicates a channel to node 6 and internally ties that channel to the channel from node 4. Node 6

completes the connection to *E*. In completing the connection, a test is made to determine if *E* is busy or is prepared to accept the connection.

2. *Data transfer.* Information can now be transmitted from A through the network to *E*. The data may be analog or digital, depending on the nature of the network. As the carriers evolve to fully integrated digital networks, the use of digital (binary) transmission for both voice and data is becoming the dominant method. The path is *A*-4 link, internal switching through 4, 4-5 channel, internal switching through 5, 5-6 channel, and internal switching through 6, 6-*E* link. Generally, the connection is full-duplex.
3. *Circuit disconnect.* After some period of data transfer, the connection is terminated, usually by the action of one of the two stations. Signals must be propagated to nodes 4, 5, and 6 to deallocate the dedicated resources.

Note that the connection path is established before data transmission begins. Thus, channel capacity must be reserved between each pair of nodes in the path, and each node must have available internal switching capacity to handle the requested connection. The switches must have the intelligence to make these allocations and to devise a route through the network.

Circuit switching can be rather inefficient. Channel capacity is dedicated for the duration of a connection, even if no data are being transferred. For a voice connection, utilization may be rather high, but it still does not approach 100 percent. For a terminal-to-computer connection, the capacity may be idle during most of the time of the connection. In terms of performance, there is a delay prior to signal transfer for call establishment. However, once the circuit is established, the network is effectively transparent to the users. Information is transmitted at a fixed data rate with no delay other than that required for propagation through the transmission links. The delay at each node is negligible.

Circuit switching was developed to handle voice traffic but is now also used for data traffic. The best-known example of a circuit-switching network is the public telephone network (Figure 8.2); this is actually a collection of national networks interconnected to form the international service. Although originally designed and implemented to service analog telephone subscribers, the network handles substantial data traffic via modem and is gradually being converted to a digital network. Another well-known application of circuit switching is the private branch exchange (PBX), used to interconnect telephones within a building or office. Circuit switching is also used in private networks—corporations or other large organizations interconnecting their various sites; these usually consist of PBX systems at each site interconnected by dedicated, leased lines obtained from one of the carriers, such as AT&T. A final common example of the application of circuit switching is the data switch. The data switch is similar to the PBX but is designed to interconnect digital data-processing devices, such as terminals and computers.

A public telecommunications network can be described using four generic architectural components:

- **Subscribers:** The devices that attach to the network. It is still the case that most subscriber devices to public telecommunications networks are telephones, but the percentage of data traffic increases year by year.

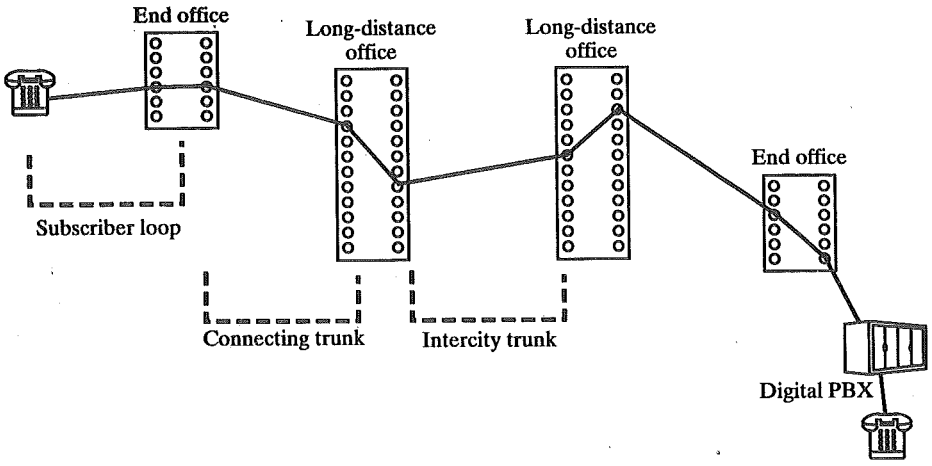


FIGURE 8.2 Public circuit-switching network.

- **Local loop:** The link between the subscriber and the network, also referred to as the *subscriber loop*. Almost all local loop connections used twisted-pair wire. The length of a local loop is typically in a range from a few kilometers to a few tens of kilometers.
- **Exchanges:** The switching centers in the network. A switching center that directly supports subscribers is known as an *end office*. Typically, an end office will support many thousands of subscribers in a localized area. There are over 19,000 end offices in the United States, so it is clearly impractical for each end office to have a direct link to each of the other end offices; this would require on the order of 2×10^8 links. Rather, intermediate switching nodes are used.
- **Trunks:** The branches between exchanges. Trunks carry multiple voice-frequency circuits using either FDM or synchronous TDM. Earlier, these were referred to as carrier systems.

Subscribers connect directly to an end office, which switches traffic between subscribers and between a subscriber and other exchanges. The other exchanges are responsible for routing and switching traffic between end offices; this distinction is shown in Figure 8.3. To connect two subscribers attached to the same end office, a circuit is set up between them in the same fashion as described before. If two subscribers connect to different end offices, a circuit between them consists of a chain of circuits through one or more intermediate offices. In the figure, a connection is established between lines *a* and *b* by simply setting up the connection through the end office. The connection between *c* and *d* is more complex. In *c*'s end office, a connection is established between line *c* and one channel on a TDM trunk to the intermediate switch. In the intermediate switch, that channel is connected to a channel on a TDM trunk to *d*'s end office. In that end office, the channel is connected to line *d*.

Circuit-switching technology has been driven by those applications that handle voice traffic. One of the key requirements for voice traffic is that there must be

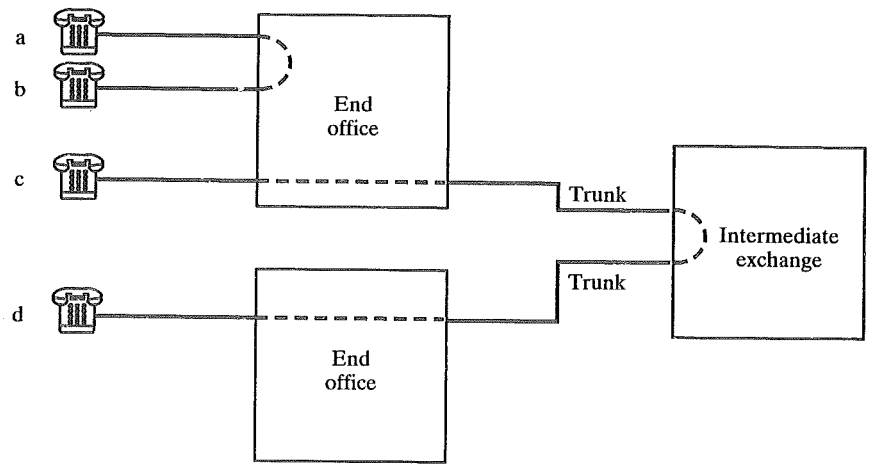


FIGURE 8.3 Circuit establishment.

virtually no transmission delay and certainly no variation in delay. A constant signal transmission rate must be maintained, as transmission and reception occur at the same signal rate. These requirements are necessary to allow normal human conversation. Further, the quality of the received signal must be sufficiently high to provide, at a minimum, intelligibility.

Circuit switching achieved its widespread, dominant position because it is well suited to the analog transmission of voice signals; in today's digital world, its inefficiencies are more apparent. However, despite the inefficiency, circuit switching will remain an attractive choice for both local-area and wide-area networking. One of its key strengths is that it is transparent. Once a circuit is established, it appears as a direct connection to the two attached stations; no special networking logic is needed at either point.

8.3 SWITCHING CONCEPTS

The technology of circuit switching is best approached by examining the operation of a single circuit-switched node. A network built around a single circuit-switching node consists of a collection of stations attached to a central switching unit. The central switch establishes a dedicated path between any two devices that wish to communicate. Figure 8.4 depicts the major elements of such a one-node network. The dotted lines inside the switch symbolize the connections that are currently active.

The heart of a modern system is a *digital switch*. The function of the digital switch is to provide a transparent signal path between any pair of attached devices. The path is transparent in that it appears to the attached pair of devices that there is a direct connection between them. Typically, the connection must allow full-duplex transmission.

The *network-interface* element represents the functions and hardware needed to connect digital devices, such as data processing devices and digital telephones, to

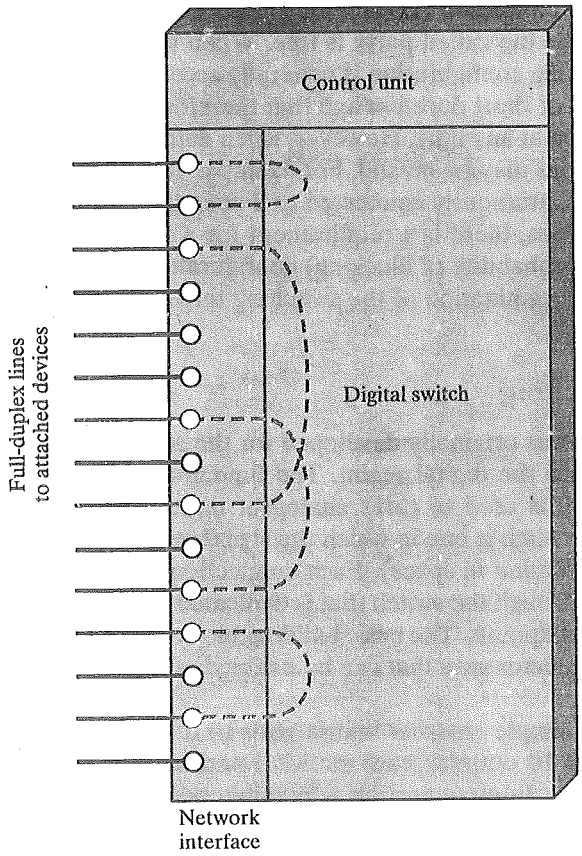


FIGURE 8.4 Elements of a circuit-switch node.

the network. Analog telephones can also be attached if the network interface contains the logic for converting to digital signals. Trunks to other digital switches carry TDM signals and provide the links for constructing multiple-node networks.

The *control unit* performs three general tasks. First, it establishes connections. This is generally done on demand—that is, at the request of an attached device. To establish the connection, the control unit must handle and acknowledge the request, determine if the intended destination is free, and construct a path through the switch. Second, the control unit must maintain the connection. Because the digital switch uses time-division principles, this may require ongoing manipulation of the switching elements. However, the bits of the communication are transferred transparently (from the point of view of the attached devices). Third, the control unit must tear down the connection, either in response to a request from one of the parties or for its own reasons.

An important characteristic of a circuit-switching device is whether it is blocking or nonblocking. Blocking occurs when the network is unable to connect two stations because all possible paths between them are already in use. A *blocking network* is one in which such blocking is possible. Hence, a *nonblocking network*

permits all stations to be connected (in pairs) at once and grants all possible connection requests as long as the called party is free. When a network is supporting only voice traffic, a blocking configuration is generally acceptable, as it is expected that most phone calls are of short duration and that therefore only a fraction of the telephones will be engaged at any time. However, when data processing devices are involved, these assumptions may be invalid. For example, for a data-entry application, a terminal may be continuously connected to a computer for hours at a time. Hence, for data applications, there is a requirement for a nonblocking or "nearly nonblocking" (very low probability of blocking) configuration.

We turn now to an examination of the switching techniques internal to a single circuit-switching node.

Space-division Switching

Space-division switching was originally developed for the analog environment and has been carried over into the digital realm. The fundamental principles are the same, whether the switch is used to carry analog or digital signals. As its name implies, a space-division switch is one in which the signal paths are physically separate from one another (divided in space). Each connection requires the establishment of a physical path through the switch that is dedicated solely to the transfer of signals between the two endpoints. The basic building block of the switch is a metallic crosspoint or semiconductor gate that can be enabled and disabled by a control unit.

Figure 8.5 shows a simple crossbar matrix with 10 full-duplex I/O lines. The matrix has 10 inputs and 10 outputs; each station attaches to the matrix via one input and one output line. Interconnection is possible between any two lines by enabling the appropriate crosspoint. Note that a total of 100 crosspoints is required. The crossbar switch has a number of limitations:

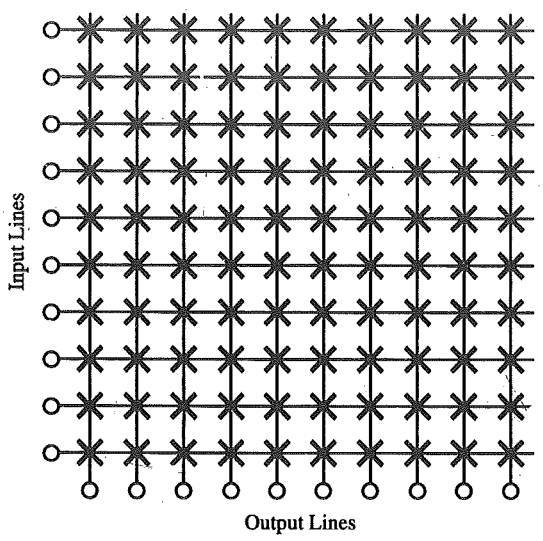


FIGURE 8.5 Space-division switch.

- The number of crosspoints grows with the square of the number of attached stations. This is costly for a large switch.
- The loss of a crosspoint prevents connection between the two devices whose lines intersect at that crosspoint.
- The crosspoints are inefficiently utilized; even when all of the attached devices are active, only a small fraction of the crosspoints are engaged.

To overcome these limitations, multiple-stage switches are employed. Figure 8.6 is an example of a three-stage switch. This type of arrangement has several advantages over a single-stage crossbar matrix:

- The number of crosspoints is reduced, increasing crossbar utilization. In this example, the total number of crosspoints for 10 stations is reduced from 100 to 48.
- There is more than one path through the network to connect two endpoints, increasing reliability.

Of course, a multistage network requires a more complex control scheme. To establish a path in a single-stage network, it is only necessary to enable a single gate. In a multistage network, a free path through the stages must be determined and the appropriate gates enabled.

A consideration with a multistage space-division switch is that it may be blocking. It should be clear from Figure 8.5 that a single-stage crossbar matrix is nonblocking; that is, a path is always available to connect an input to an output; that this may not be the case with a multiple-stage switch can be seen in Figure 8.6. The

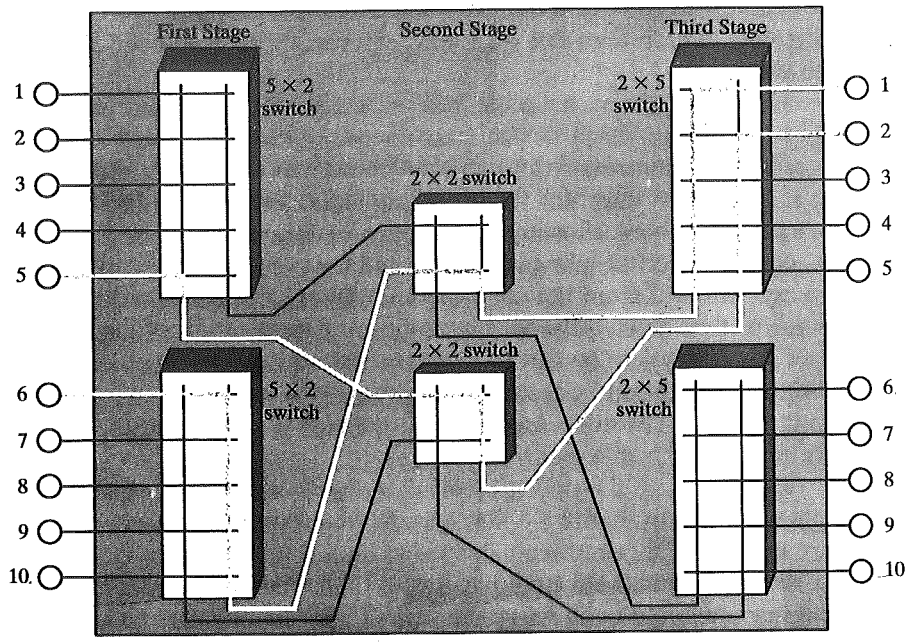


FIGURE 8.6 Three-stage space-division switch.

heavier lines indicate ones already in use. In this state, input line 10, for example, cannot be connected to output line 3, 4, or 5, even though all of these output lines are available. A multiple-stage switch can be made nonblocking by increasing the number or size of the intermediate switches, but of course this increases the cost.

Time-division Switching

The technology of switching has a long history, most of it covering an era when analog signal switching predominated. With the advent of digitized voice and synchronous time-division multiplexing techniques, both voice and data can be transmitted via digital signals; this has led to a fundamental change in the design and technology of switching systems. Instead of relatively dumb space-division systems, modern digital systems rely on intelligent control of space- and time-division elements.

Virtually all modern circuit switches use digital time-division techniques for establishing and maintaining "circuits." Time-division switching involves the partitioning of a lower-speed bit stream into pieces that share a higher-speed stream with other bit streams. The individual pieces, or slots, are manipulated by control logic to route data from input to output. There are a number of variations on this basic concept. To give the reader some feel for time-division switching, we examine one of the simplest but most popular techniques, referred to as TDM bus switching.

TDM bus switching, and indeed all digital switching techniques, are based on the use of synchronous time-division multiplexing (TDM). As we saw in Figure 7.6, synchronous TDM permits multiple low-speed bit streams to share a high-speed line. A set of inputs is sampled in turn. The samples are organized serially into slots (channels) to form a recurring frame of slots, with the number of slots per frame equal to the number of inputs. A slot may be a bit, a byte, or some longer block. An important point to note is that with synchronous TDM, the source and destination of the data in each time slot are known. Hence, there is no need for address bits in each slot.

Figure 8.7 shows a simple way in which this technique can be adapted to achieve switching. Each device attaches to the switch through a full-duplex line. These lines are connected through controlled gates to a high-speed digital bus. Each line is assigned a time slot for providing input. For the duration of the slot, that line's gate is enabled, allowing a small burst of data onto the bus. For that same time slot, one of the other line gates is enabled for output. Thus, during that time slot, data are switched from the enabled input line to the enabled output line. During successive time slots, different input/output pairings are enabled, allowing a number of connections to be carried over the shared bus. An attached device achieves full-duplex operation by transmitting during one assigned time slot and receiving during another. The other end of the connection is an I/O pair for which these time slots have the opposite meanings.

Let us look at the timing involved more closely. First, consider a nonblocking implementation of Figure 8.7. For a switch that supports, for example, 100 devices, there must be 100 repetitively occurring time slots, each one assigned to an input and an output line. One iteration for all time slots is referred to as a *frame*. The input assignment may be fixed; the output assignments vary to allow various connections. When a time slot begins, the designated (enabled) input line may insert a burst of data onto the line, where it will propagate to both ends past all other lines.

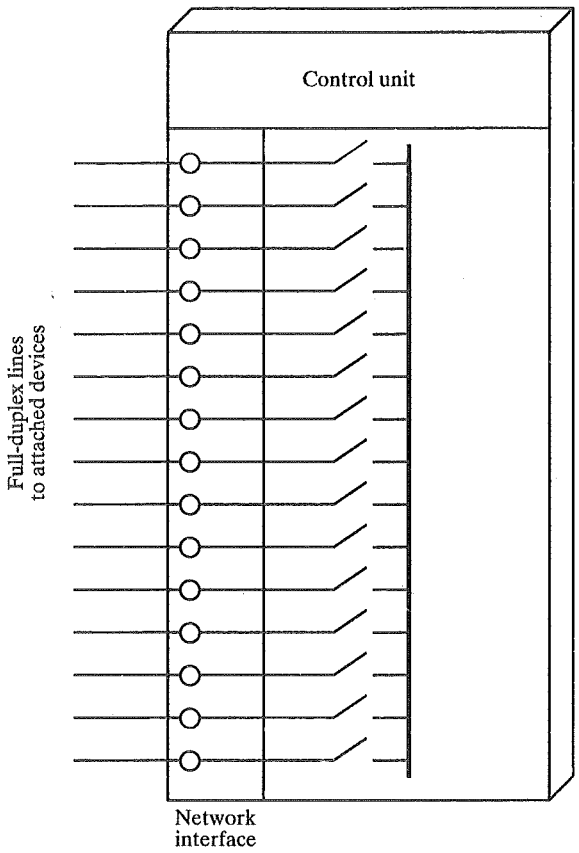


FIGURE 8.7 TDM bus switching.

The designated (enabled) output line, during that time, copies the data, if present, as they go by. The time slot, therefore, must equal the transmission time of the input plus the propagation delay between input and output across the bus. In order to keep successive time slots uniform, time-slot length is defined as transmission time plus the end-to-end bus propagation delay.

To keep up with the input lines, the data rate on the bus must be high enough that the slots recur sufficiently frequently. For example, consider a system connecting 100 full-duplex lines at 19.2 kbps. Input data on each line are buffered at the gate. Each buffer must be cleared, by enabling the gate, quickly enough to avoid overrun. Thus, the data rate on the bus in this example must be greater than 1.92 Mbps. The actual data rate must be high enough to also account for the wasted time due to propagation delay.

The above considerations determine the traffic-carrying capacity of a blocking switch, as well, where there is no fixed assignment of input lines to time slots; they are allocated on demand. The data rate on the bus dictates how many connections can be made at a time. For a system with 200 devices at 19.2 kbps and a bus at 2 Mbps, about half of the devices can be connected at any one time.

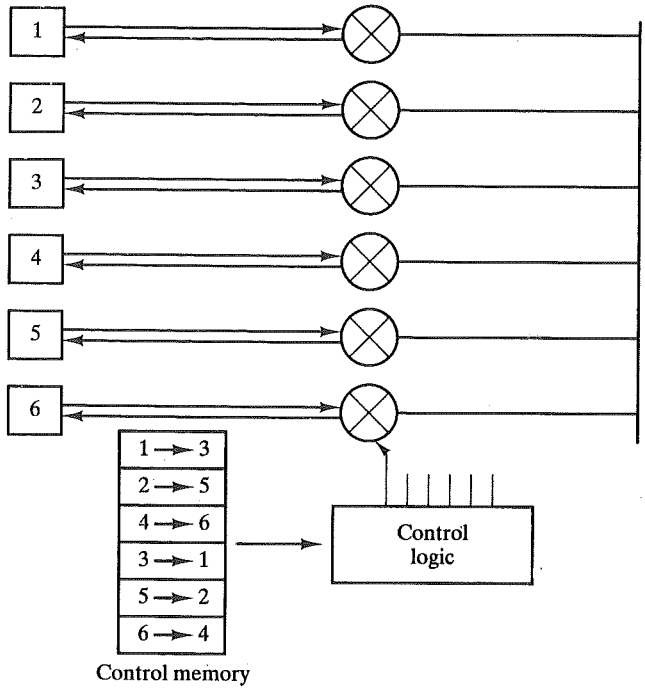


FIGURE 8.8 Control of a TDM bus switch.

The TDM bus-switching scheme can accommodate lines of varying data rates. For example, if a 9600-bps line gets one slot per frame, a 19.2-kbps line would get two slots per frame. Of course, only lines of the same data rate can be connected.

Figure 8.8 is an example that suggests how the control for a TDM bus switch can be implemented. Let us assume that the propagation time on the bus is 0.01 μsec. Time on the bus is organized into 30.06-μsec frames of six 5.01-μsec time slots each. A control memory indicates which gates are to be enabled during each time slot. In this example, six words of memory are needed. A controller cycles through the memory at a rate of one cycle every 30.06 μsec. During the first time slot of each cycle, the input gate from device 1 and the output gate to device 3 are enabled, allowing data to pass from device 1 to device 3 over the bus. The remaining words are accessed in succeeding time slots and treated accordingly. As long as the control memory contains the contents depicted in Figure 8.8, connections are maintained between 1 and 3, 2 and 5, and 4 and 6.

8.4 ROUTING IN CIRCUIT-SWITCHED NETWORKS

In a large circuit-switched network, such as the AT&T long-distance telephone network, many of the circuit connections will require a path through more than one

switch. When a call is placed, the network must devise a route through the network from calling subscriber to called subscriber that passes through some number of switches and trunks. There are two main requirements for the network's architecture that bear on the routing strategy: efficiency and resilience. First, it is desirable to minimize the amount of equipment (switches and trunks) in the network subject to the ability to handle the expected load. The load requirement is usually expressed in terms of a *busy-hour traffic load*; this is simply the average load expected over the course of the busiest hour of use during the course of a day. From a functional point of view, it is necessary to handle that amount of load. From a cost point of view, we would like to handle that load with minimum equipment. However, there is another requirement, namely, resilience. Although the network may be sized for the busy hour load, it is possible for the traffic to temporarily surge above that level (for example, during a major storm). It will also be the case that, from time to time, switches and trunks will fail and be temporarily unavailable (unfortunately, maybe during the same storm). We would like the network to provide a reasonable level of service under such conditions.

The key design issue that determines the nature of the tradeoff between efficiency and resilience is the routing strategy. Traditionally, the routing function in public telecommunications networks has been quite simple. In essence, the switches of a network were organized into a tree structure, or hierarchy. A path was constructed by starting at the calling subscriber, tracing up the tree to the first common node, and then tracing down the tree to the called subscriber. To add some resilience to the network, additional high-usage trunks were added that cut across the tree structure to connect exchanges with high volumes of traffic between them; in general, this is a static approach. The addition of high-usage trunks provides redundancy and extra capacity, but limitations remain both in efficiency and resilience. Because this routing scheme is not able to adapt to changing conditions, the network must be designed to meet some typical heavy demand. As an example of the problems raised by this approach, the busy hours for east-west traffic and those for north-south traffic do not coincide; they each place different demands on the system. It is difficult to analyze the effects of these variables, which leads to oversizing and, ultimately, inefficiency. In terms of resilience, the fixed hierarchical structure with supplemental trunks may respond poorly to failures. Typically in such designs, the result of a failure is a major local congestion at that location.

To cope with the growing demands on public telecommunications networks, virtually all providers have moved away from the static hierarchical approach to a dynamic approach. A dynamic routing approach is one in which routing decisions are influenced by current traffic conditions. Typically, the circuit-switching nodes have a peer relationship with each other rather than a hierarchical one. All nodes are capable of performing the same functions. In such an architecture, routing is both more complex and more flexible. It is more complex because the architecture does not provide a "natural" path or set of paths based on hierarchical structure; but it is also more flexible, as more alternative routes are available.

Two broad classes of dynamic routing algorithms have been implemented: alternate routing and adaptive routing.

Alternate Routing

The essence of alternate-routing schemes is that the possible routes to be used between two end offices are predefined. It is the responsibility of the originating switch to select the appropriate route for each call. Each switch is given a set of preplanned routes for each destination, in order of preference. The preferred choice is a direct trunk connection between two switches. If this trunk is unavailable, then the second choice is to be tried, and so on. The routing sequences (sequence in which the routes in the set are tried) reflect an analysis based on historical traffic patterns, and are designed to optimize the use of network resources.

If there is only one routing sequence defined for each source-destination pair, the scheme is known as a fixed alternate-routing scheme. More commonly, a dynamic alternate-routing scheme is used. In the latter case, a different set of preplanned routes is used for different time periods, to take advantage of the differing traffic patterns in different time zones and at different times of day. Thus, the routing decision is based both on current traffic status (a route is rejected if busy) and historical traffic patterns (which determine the sequence of routes to be considered).

A simple example is shown in Figure 8.9. The originating switch, X, has four possible routes to the destination switch, Y. The direct route (a) will always be tried first. If this trunk is unavailable (busy, out of service), the other routes will be tried in a particular order, depending on the time period. For example, during weekday mornings, route b is tried next.

A form of the dynamic alternate-routing technique is employed by the Bell Operating Companies for providing local and regional telephone service [BELL90]; it is referred to as multialternate routing (MAR). This approach is also used by AT&T in its long-distance network [ASH90], and is referred to as dynamic nonhierarchical routing (DNHR).

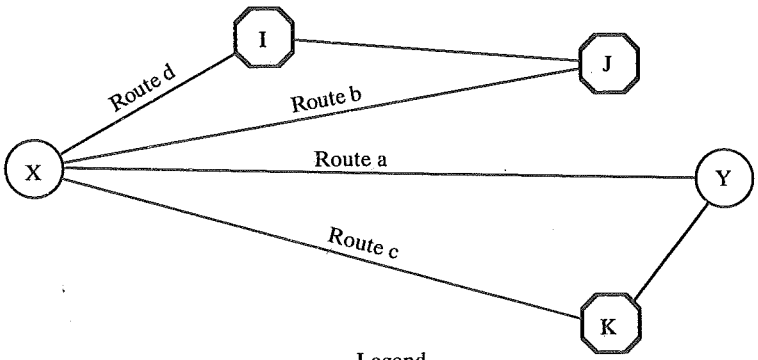
Adaptive Routing

An adaptive-routing scheme is designed to enable switches to react to changing traffic patterns on the network. Such schemes require greater management overhead, as the switches must exchange information to learn of network conditions. However, compared to an alternate-routing scheme, an adaptive scheme has the potential for more effectively optimizing the use of network resources. In this subsection, we briefly describe an important example of an adaptive-routing scheme.

Dynamic traffic management (DTM) is a routing capability developed by Northern Telecom and used in the Canadian national and local telephone networks [REGN90].

DTM uses a central controller to find the best alternate route choices depending on congestion in the network. The central controller collects status data from each switch in the network every 10 seconds to determine preferred alternate routes. Each call is first attempted on the direct path, if any exists, between source and destination switches. If the call is blocked, it is attempted on a two-link alternate path.

Each switch i communicates the following traffic measurements to the central controller:



Route a: X→ Y
Route b: X→ J → Y
Route c: X→ K→ Y
Route d: X→ I → J → Y

(a) Topology

Time Period	First route	Second route	Third route	Fourth and final route
Morning	a	b	c	d
Afternoon	a	d	b	c
Evening	a	d	c	b
Weekend	a	c	b	d

(b) Routing table

FIGURE 8.9 Alternate routes from end office X to end office.

I_{ij} = The number of idle trunks on the link to switch j, for all switches in the network

CPU_i = The CPU utilization of switch i

O_{ij} = A measure of the traffic sent by i to j that overflowed the direct route.

Based on this information, the central controller periodically returns to each switch i, for each possible destination switch j:

r_{ij} = The identifier of the switch through which i should direct its calls to j when the direct link is full.

The selection of r_{ij} depends on whether or not a direct link exists between i and j. If a direct link exists, which is the case for the vast majority of the calls, then r_{ij} is determined as that switch t that achieves the maximum in

$$\text{Max } \{A_t \times \text{Min } [I_{it} - PA_{it}, I_{tj} - PA_{tj}]\} \quad t \neq i,j$$

If there is no direct link between i and j, then r_{ij} is determined as that switch t that achieves the maximum in

$$\text{Max } \{A_t \times \text{Min } [I_{it}, I_{tj}]\} \qquad t \neq i,j$$

where

A_t = Parameter in the range [0,1] that reflects the availability of switch t. It is 1 if t functions normally, but it is less if t is overloaded; its role is to make alternative routes that transit through overloaded switches less attractive and, hence, less likely to be chosen by the network controller.

PA_{xy} = Protective-allowance parameter for the direct traffic on link x-y; its role is to divert traffic away from the link when it is nearly fully occupied.

The second equation is the same as the first, except that protective allowances are not considered. The rationale is the following: If there is no direct link between switches i and j, then traffic from i to j should not concede priority to direct traffic over links on potential alternate routes.

Figure 8.10 illustrates the selection process. If the link from i to j is saturated, the recommended alternate route is i-y-j. Although route i-x-j has the largest idle capacity, it is not recommended because switch x is overloaded.

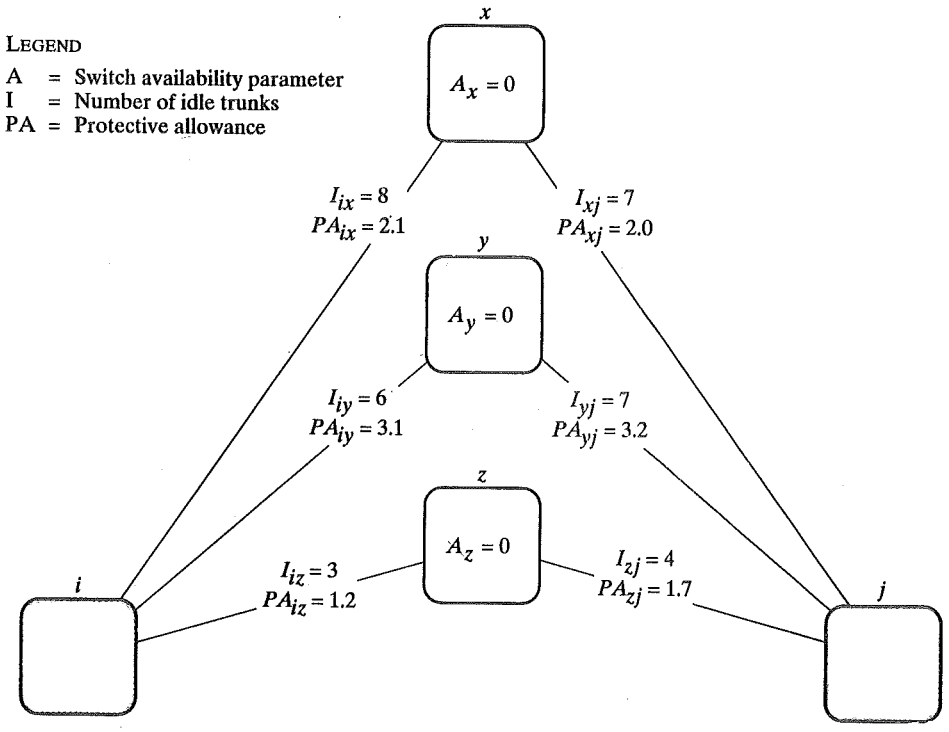


FIGURE 8.10 Adaptive route selection in DTM.

The use of a set of parameters based on network status provides a powerful routing capability. Furthermore, it becomes an easy matter to experiment with various ways of determining the values of parameters and assessing their effect on performance. For example, the parameter PA_{xy} can be set to a fixed value in a relatively stable network, or the overflow measurement O_{xy} can be used.

8.5 CONTROL SIGNALING

In a circuit-switched network, control signals are the means by which the network is managed and by which calls are established, maintained, and terminated. Both call management and overall network management require that information be exchanged between subscriber and switch, among switches, and between switch and network management center. For a large public telecommunications network, a relatively complex control-signaling scheme is required. In this section, we provide a brief overview of control-signal functionality and then look at the technique that is the basis of modern integrated digital networks: common channel signaling.

Signaling Functions

Control signals affect many aspects of network behavior, including both network services visible to the subscriber and internal mechanisms. As networks become more complex, the number of functions performed by control signaling necessarily grows. The following functions, listed in [MART90], are among the most important:

1. Audible communication with the subscriber, including dial tone, ringing tone, busy signal, and so on.
2. Transmission of the number dialed to switching offices that will attempt to complete a connection.
3. Transmission of information between switches indicating that a call cannot be completed.
4. Transmission of information between switches indicating that a call has ended and that the path can be disconnected.
5. A signal to make a telephone ring.
6. Transmission of information used for billing purposes.
7. Transmission of information giving the status of equipment or trunks in the network. This information may be used for routing and maintenance purposes.
8. Transmission of information used in diagnosing and isolating system failures.
9. Control of special equipment such as satellite channel equipment.

As an example of the use of control signaling, consider a typical telephone connection sequence from one line to another in the same central office:

1. Prior to the call, both telephones are not in use (on-hook). The call begins when one subscriber lifts the receiver (off-hook); this action is automatically signaled to the end office switch.
2. The switch responds with an audible dial tone, signaling the subscriber that the number may be dialed.
3. The caller dials the number, which is communicated as a called address to the switch.
4. If the called subscriber is not busy, the switch alerts that subscriber to an incoming call by sending a ringing signal, which causes the telephone to ring.
5. Feedback is provided to the calling subscriber by the switch:
   a) If the called subscriber is not busy, the switch returns an audible ringing tone to the caller while the ringing signal is being sent to the called subscriber.
   b) If the called subscriber is busy, the switch sends an audible busy signal to the caller.
   c) If the call cannot be completed through the switch, the switch sends an audible "reorder" message to the caller.
6. The called party accepts the call by lifting the receiver (off-hook), which is automatically signaled to the switch.
7. The switch terminates the ringing signal and the audible ringing tone, and establishes a connection between the two subscribers.
8. The connection is released when either subscriber hangs up.

When the called subscriber is attached to a different switch than that of the calling subscriber, the following switch-to-switch trunk signaling functions are required:

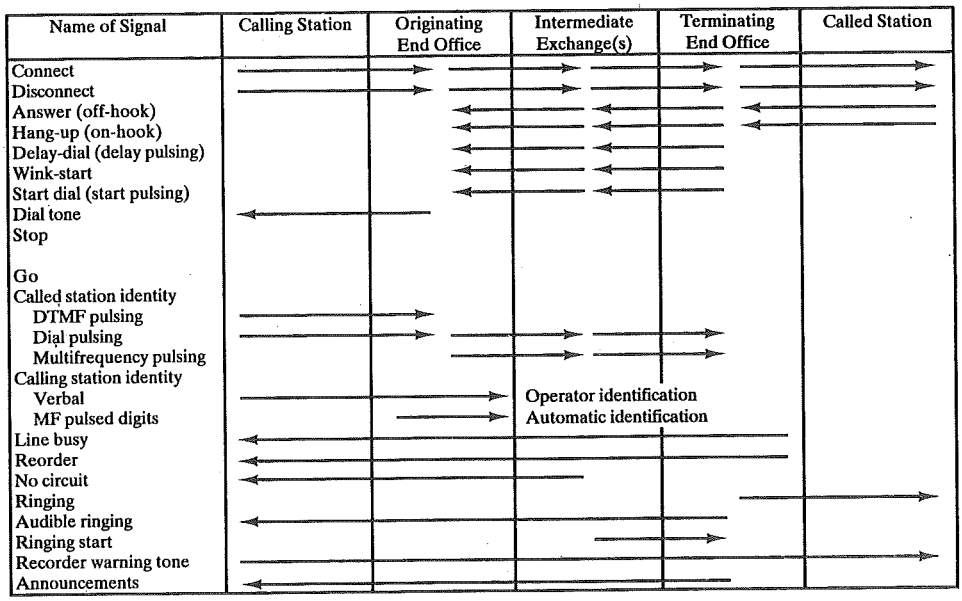
1. The originating switch seizes an idle interswitch trunk, sends an off-hook indication on the trunk, and requests a digit register at the far end, so that the address may be communicated.
2. The terminating switch sends an off-hook followed by an on-hook signal, known as a "wink." This indicates a register-ready status.
3. The originating switch sends the address digits to the terminating switch.

This example illustrates some of the functions performed using control signals.

Figure 8.11, based on a presentation in [FREE94], indicates the origin and destination of various control signals. Signaling can also be classified functionally as supervisory, address, call-information, and network-management.

The term *supervisory* is generally used to refer to control functions that have a binary character (true/false; on/off), such as request for service, answer, alerting, and return to idle; they deal with the availability of the called subscriber and of the needed network resources. Supervisory control signals are used to determine if a needed resource is available and, if so, to seize it; they are also used to communicate the status of requested resources.

Address signals identify a subscriber. Initially, an address signal is generated by a calling subscriber when dialing a telephone number. The resulting address may



Note: A broken line indicates repetition of a signal at each office, whereas a solid line indicates direct transmittal through intermediate offices.

FIGURE 8.11 Control signaling through a circuit-switched telephone network.

be propagated through the network to support the routing function and to locate and ring the called subscriber's phone.

The term *call-information* refers to those signals that provide information to the subscriber about the status of a call. This is in contrast to internal control signals between switches used in call establishment and termination. Such internal signals are analog or digital electrical messages. In contrast, call information signals are audible tones that can be heard by the caller or an operator with the proper phone set.

Supervisory, address, and call-information control signals are directly involved in the establishment and termination of a call. In contrast, *network-management* signals are used for the maintenance, troubleshooting, and overall operation of the network. Such signals may be in the form of messages, such as a list of preplanned routes being sent to a station to update its routing tables. These signals cover a broad scope, and it is this category that will expand most with the increasing complexity of switched networks.

Location of Signaling

Control signaling needs to be considered in two contexts: signaling between a subscriber and the network, and signaling within the network. Typically, signaling operates differently within these two contexts.

The signaling between a telephone or other subscriber device and the switching office to which it attaches is, to a large extent, determined by the characteristics

of the subscriber device and the needs of the human user. Signals within the network are entirely computer-to-computer. The internal signaling is concerned not only with the management of subscriber calls but with the management of the network itself. Thus, for internal signaling, a more complex repertoire of commands, responses, and set of parameters is needed.

Because two different signaling techniques are used, the local switching office to which the subscriber is attached must provide a mapping between the relatively less complex signaling technique used by the subscriber and the more complex technique used within the network.

Common Channel Signaling

Traditional control signaling in circuit-switched networks has been on a per-trunk or inchannel basis. With *inchannel signaling*, the same channel is used to carry control signals as is used to carry the call to which the control signals relate. Such signaling begins at the originating subscriber and follows the same path as the call itself. This process has the merit that no additional transmission facilities are needed for signaling; the facilities for voice transmission are shared with control signaling.

Two forms of inchannel signaling are in use: inband and out-of-band. *Inband signaling* uses not only the same physical path as the call it serves; it also uses the same frequency band as the voice signals that are carried. This form of signaling has several advantages. Because the control signals have the same electromagnetic properties as the voice signals, they can go anywhere that the voice signals go. Thus, there are no limits on the use of inband signaling anywhere in the network, including places where analog-to-digital or digital-to-analog conversion takes place. In addition, it is impossible to set up a call on a faulty speech path, as the control signals that are used to set up that path would have to follow the same path.

Out-of-band signaling takes advantage of the fact that voice signals do not use the full 4-kHz bandwidth allotted to them. A separate narrow signaling band within the 4 kHZ is used to send control signals. The major advantage of this approach is that the control signals can be sent whether or not voice signals are on the line, thus allowing continuous supervision and control of a call. However, an out-of-band scheme needs extra electronics to handle the signaling band, and the signaling rates are slower because the signal has been confined to a narrow bandwidth.

As public telecommunications networks become more complex and provide a richer set of services, the drawbacks of inchannel signaling become more apparent. The information transfer rate is quite limited with inchannel signaling. With inband signals, the voice channel being used is only available for control signals when there are no voice signals on the circuit. With out-of-band signals, a very narrow bandwidth is available. With such limits, it is difficult to accommodate, in a timely fashion, any but the simplest form of control messages. However, to take advantage of the potential services and to cope with the increasing complexity of evolving network technology, a richer and more powerful control signal repertoire is needed.

A second drawback of inchannel signaling is the amount of delay from the time a subscriber enters an address (dials a number) to when the connection is established. The requirement to reduce this delay is becoming more important as the network is used in new ways. For example, computer-controlled calls, such as

with transaction processing, use relatively short messages; therefore, the call setup time represents an appreciable part of the total transaction time.

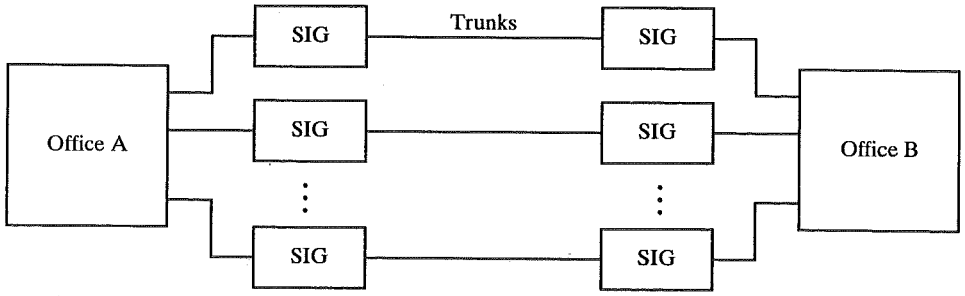
Both of these problems can be addressed with *common-channel signaling*, in which control signals are carried over paths completely independent of the voice channels (Table 8.1). One independent control signal path can carry the signals for a number of subscriber channels, and, hence, is a common control channel for these subscriber channels.

TABLE 8.1 Signaling techniques for circuit-switched networks.

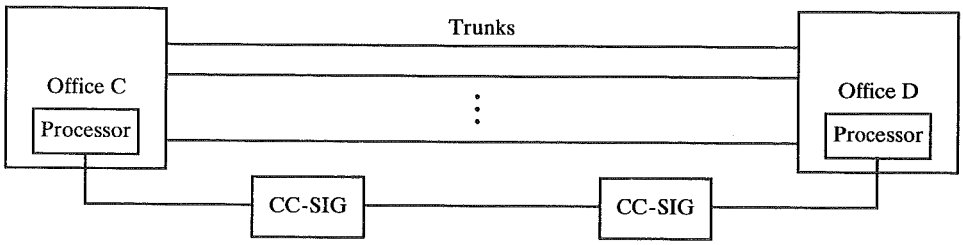
	Description	Comment
Inchannel		
Inband	Transmit control signals in the same band of frequencies used by the voice signals.	The simplest technique. It is necessary for call information signals and may be used for other control signals. Inband can be used over any type of subscriber line interface.
Out-of-band	Transmit control signals over the same facilities as the voice signal but a different part of the frequency band.	Unlike inband, out-of-band provides continuous supervision for the duration of a connection.
Common Channel	Transmit control signals over signaling channels that are dedicated to control signals and are common to a number of voice channels.	Reduces call setup time compared with inchannel methods. It is also more adaptable to evolving functional needs.

The principle of common-channel signaling is illustrated and contrasted with inchannel signaling in Figure 8.12. As can be seen, the signal path for common-channel signaling is physically separate from the path for voice or other subscriber signals. The common channel can be configured with the bandwidth required to carry control signals for a rich variety of functions. Thus, both the signaling protocol and the network architecture to support that protocol are more complex than inchannel signaling. However, the continuing drop in computer hardware costs makes common-channel signaling increasingly attractive. The control signals are messages that are passed between switches as well as between a switch and the network management center. Thus, the control-signaling portion of the network is, in effect, a distributed computer network carrying short messages.

Two modes of operation are used in common-channel signaling (Figure 8.13). In the *associated mode,* the common channel closely tracks along its entire length the interswitch trunk groups that are served between endpoints. The control signals are on different channels from the subscriber signals, and, inside the switch, the control signals are routed directly to a control signal processor. A more complex, but more powerful, mode is the *nonassociated mode*; with this, the network is augmented by additional nodes, known as signal transfer points. There is now no close or simple assignment of control channels to trunk groups. In effect, there are now two separate networks, with links between them so that the control portion of the



(a) Inchannel



(b) Common channel

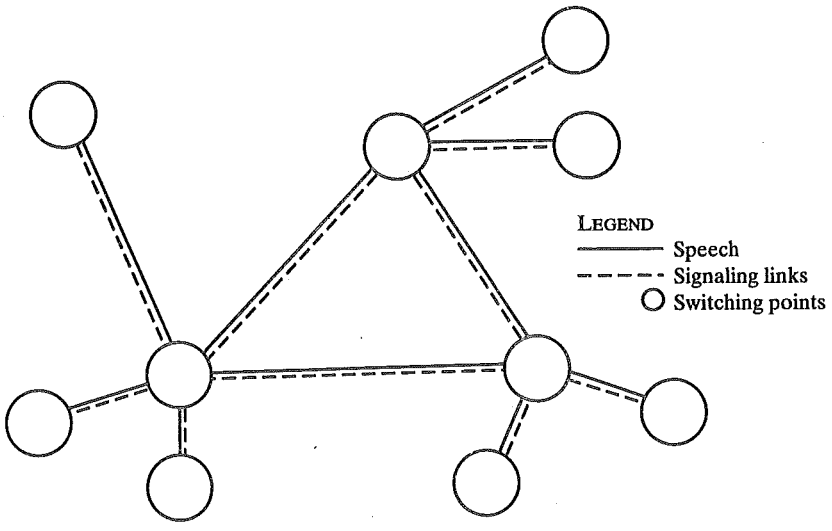
LEGEND
SIG = Per-trunk signaling equipment
CC-SIG = Common-channel signaling equipment

FIGURE 8.12 Inchannel and common-channel signaling.

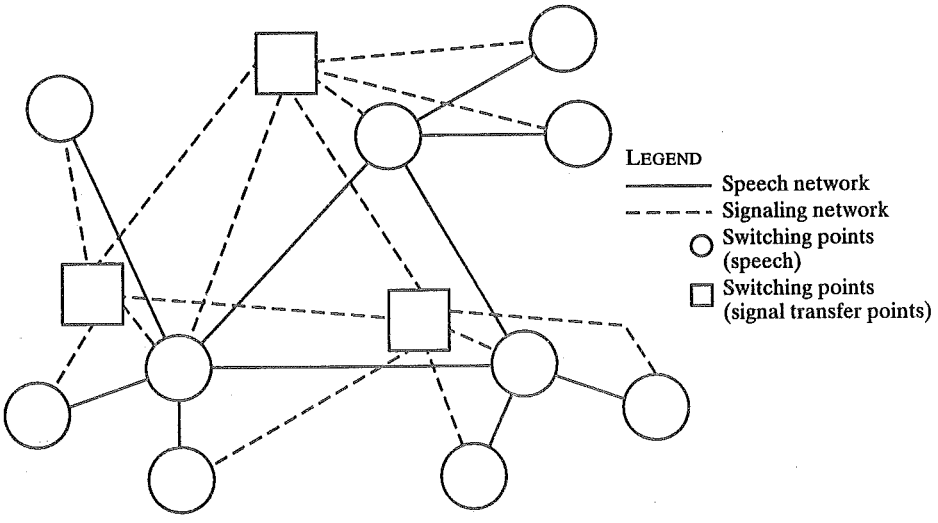
network can exercise control over the switching nodes that are servicing the subscriber calls. Network management is more easily exerted in the nonassociated mode as control channels can be assigned to tasks in a more flexible manner. The nonassociated mode is the mode used in ISDN.

With inchannel signaling, control signals from one switch are originated by a control processor and switched onto the outgoing channel. On the receiving end, the control signals must be switched from the voice channel into the control processor. With common-channel signaling, the control signals are transferred directly from one control processor to another, without being tied to a voice signal; this is a simpler procedure, and one of the main motivations for common-channel signaling as it is less susceptible to accidental or intentional interference between subscriber and control signals. Another key motivation for common-channel signaling is that call-setup time is reduced. Consider the sequence of events for call setup with inchannel signaling when more than one switch is involved. A control signal will be sent from one switch to the next in the intended path. At each switch, the control signal cannot be transferred through the switch to the next leg of the route until the associated circuit is established through that switch. With common-channel signaling, forwarding of control information can overlap the circuit-setup process.

With nonassociated signaling, a further advantage emerges: One or more central control points can be established. All control information can be routed to a



(a) Associated



(b) Nonassociated

FIGURE 8.13 Common-channel signaling modes.

network control center where requests are processed and from which control signals are sent to switches that handle subscriber traffic; in this way, requests can be processed with a more global view of network conditions.

Of course, there are disadvantages to common-channel signaling; these primarily have to do with the complexity of the technique. However, the dropping cost of digital hardware and the increasingly digital nature of telecommunication networks makes common-channel signaling the appropriate technology.

All of the discussion in this section has dealt with the use of common-channel signaling inside the network—that is, to control switches. Even in a network that is completely controlled by common-channel signaling, inchannel signaling is needed for at least some of the communication with the subscriber. For example, dial tone, ringback, and busy signals must be inchannel to reach the user. In a simple telephone network, the subscriber does not have access to the common-channel signaling portion of the network and does not employ the common-channel signaling protocol. However, in more sophisticated digital networks, including ISDN, a common-channel signaling protocol is employed between subscriber and network, and is mapped to the internal-signaling protocol.

8.6 RECOMMENDED READING

As befits its age, circuit switching has inspired a voluminous literature. Two good books on the subject are [BELL91] and [FREE96]. [MART90] also has a highly readable treatment.

The October 1990 issue of IEEE Communications magazine is devoted to the topic of routing in circuit-switched networks. [GIRA90] provides good coverage. Discussions of control signaling can be found in [FREE96] and [FREE94].

BELL91 Bellamy, J. *Digital Telephony*. New York: Wiley, 1991.

FREE96 Freeman, R. *Telecommunication System Engineering*. New York: Wiley, 1996.

FREE94 Freeman, R. *Reference Manual for Telecommunications Engineering*. New York: Wiley, 1994.

GIRA90 Girard, A. *Routing and Dimensioning in Circuit-Switched Networks*. Reading, MA: Addison-Wesley, 1990.

MART90 Martin, J. *Telecommunications and the Computer*. Englewood Cliffs, NJ: Prentice Hall, 1990.

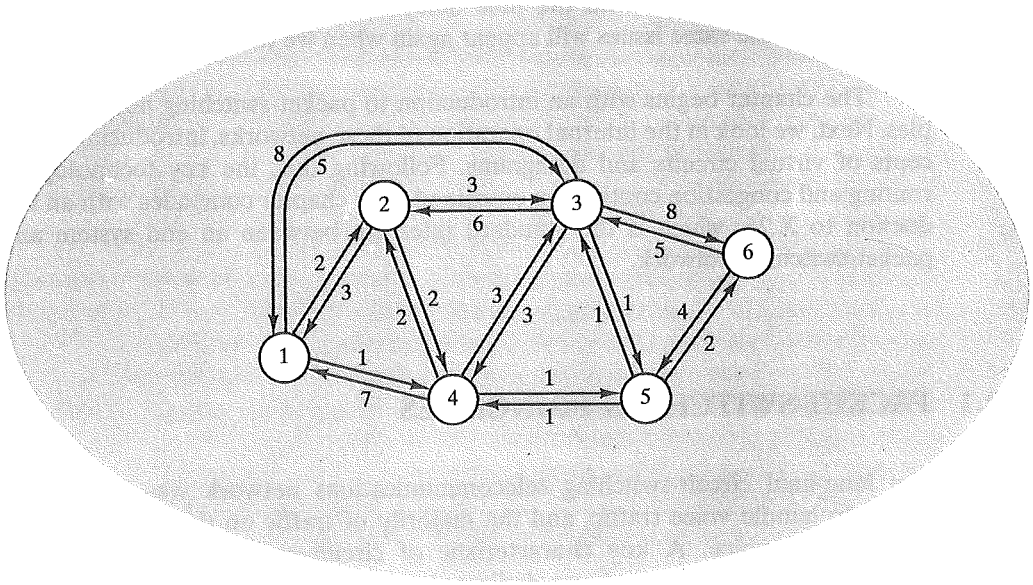
8.7 PROBLEMS

8.1 Assume that the velocity of propagation on a TDM bus is 0.8 c, its length is 10 m, and the data rate is 500 Mbps. How many bits should be transmitted in a time slot to achieve a bus efficiency of 99%?

8.2 Consider a simple telephone network consisting of two end offices and one intermediate switch with a 1-MHz full-duplex trunk between each end office and the intermediate switch. The average telephone is used to make four calls per 8-hour workday, with a mean call duration of six minutes. Ten percent of the calls are long distance, What is the maximum number of telephones an end office can support?

CHAPTER 9

PACKET SWITCHING



Around 1970, research began on a new form of architecture for long-distance digital data communications: packet switching. Although the technology of packet switching has evolved substantially since that time, it is remarkable that (1) the basic technology of packet switching is fundamentally the same today as it was in the early-1970s networks, and (2) packet switching remains one of the few effective technologies for long-distance data communications.

This chapter provides an overview of packet-switching technology. We will see that many of the advantages of packet switching (flexibility, resource sharing, robustness, responsiveness) come with a cost. The packet-switching network is a distributed collection of packet-switching nodes. Ideally, all packet-switching nodes would always know the state of the entire network. Unfortunately, because the nodes are distributed, there is always a time delay between a change in status in one portion of the network and the knowledge of that change elsewhere. Furthermore, there is overhead involved in communicating status information. As a result, a packet-switching network can never perform "perfectly," and so elaborate algorithms are used to cope with the time delay and overhead penalties of network operation. These same issues will appear again when we discuss internetworking in Part IV.

The chapter begins with an introduction to packet-switching network principles. Next, we look at the internal operation of these networks, introducing the concepts of virtual circuits and datagrams. Following this, the key technologies of routing and congestion control are examined. The chapter concludes with an introduction to X.25, which is the standard interface between an end system and a packet-switching network.

9.1 PACKET-SWITCHING PRINCIPLES

The long-haul circuit-switching telecommunications network was orginally designed to handle voice traffic, and the majority of traffic on these networks continues to be voice. A key characteristic of circuit-switching networks is that resources within the network are dedicated to a particular call. For voice connections, the resulting circuit will enjoy a high percentage of utilization because, most of the time, one party or the other is talking. However, as the circuit-switching network began to be used increasingly for data connections, two shortcomings became apparent:

- In a typical user/host data connection (e.g., personal computer user logged on to a database server), much of the time the line is idle. Thus, with data connections, a circuit-switching approach is inefficient.
- In a circuit-switching network, the connection provides for transmission at constant data rate. Thus, each of the two devices that are connected must transmit and receive at the same data rate as the other; this limits the utility of the network in interconnecting a variety of host computers and terminals.

To understand how packet switching addresses these problems, let us briefly summarize packet-switching operation. Data are transmitted in short packets. A typical upper bound on packet length is 1000 octets (bytes). If a source has a longer message to send, the message is broken up into a series of packets (Figure 9.1). Each packet contains a portion (or all for a short message) of the user's data plus some control information. The control information, at a minimum, includes the information that the network requires in order to be able to route the packet through the network and deliver it to the intended destination. At each node en route, the packet is received, stored briefly, and passed on to the next node.

Let us return to Figure 8.1, but now assume that it depicts a simple packet-switching network. Consider a packet to be sent from station *A* to station *E*. The packet will include control information that indicates that the intended destination is *E*. The packet is sent from *A* to node 4. Node 4 stores the packet, determines the next leg of the route (say 5), and queues the packet to go out on that link (the 4-5 link). When the link is available, the packet is transmitted to node 5, which will forward the packet to node 6, and finally to *E*. This approach has a number of advantages over circuit switching:

- Line efficiency is greater, as a single node-to-node link can be dynamically shared by many packets over time. The packets are queued up and transmitted as rapidly as possible over the link. By contrast, with circuit switching, time on a node-to-node link is preallocated using synchronous time-division multiplexing. Much of the time, such a link may be idle because a portion of its time is dedicated to a connection which is idle.
- A packet-switching network can perform data-rate conversion. Two stations of different data rates can exchange packets because each connects to its node at its proper data rate.

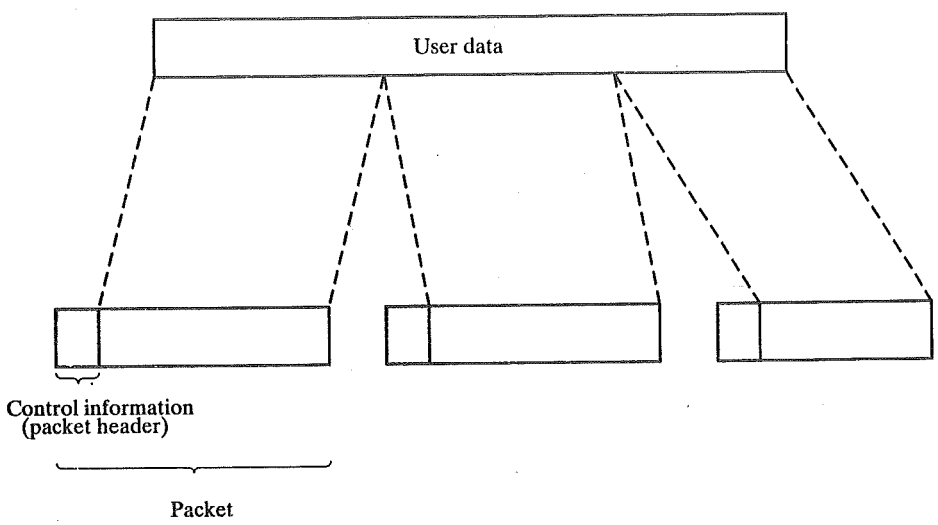


FIGURE 9.1 Packets.

- When traffic becomes heavy on a circuit-switching network, some calls are blocked; that is, the network refuses to accept additional connection requests until the load on the network decreases. On a packet-switching network, packets are still accepted, but delivery delay increases.
- Priorities can be used. Thus, if a node has a number of packets queued for transmission, it can transmit the higher-priority packets first. These packets will therefore experience less delay than lower-priority packets.

Switching Technique

A station has a message to send through a packet-switching network that is of length greater than the maximum packet size. It therefore breaks the message up into packets and sends these packets, one at a time, to the network. A question arises as to how the network will handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination; there are two approaches that are used in contemporary networks: datagram and virtual circuit.

In the datagram approach, each packet is treated independently, with no reference to packets that have gone before. Let us consider the implication of this approach. Suppose that station *A* in Figure 8.1 has a three-packet message to send to E. It transmits the packets, 1-2-3, to node 4. On each packet, node 4 must make a routing decision. Packet 1 arrives for delivery to *E*. Node 4 could plausibly forward this packet to either node 5 or node 7 as the next step in the route. In this case, node 4 determines that its queue of packets for node 5 is shorter than for node 7, so it queues the packet for node 5. Ditto for packet 2. But for packet 3, node 4 finds that its queue for node 7 is now shorter and so queues packet 3 for that node. So the packets, each with the same destination address, do not all follow the same route. As a result, it is possible that packet 3 will beat packet 2 to node 6. Thus, it is also possible that the packets will be delivered to *E* in a different sequence from the one in which they were sent. It is up to *E* to figure out how to reorder them. Also, it is possible for a packet to be destroyed in the network. For example, if a packet-switching node crashes momentarily, all of its queued packets may be lost. If this were to happen to one of the packets in our example, node 6 has no way of knowing that one of the packets in the sequence of packets has been lost. Again, it is up to *E* to detect the loss of a packet and figure out how to recover it. In this technique, each packet, treated independently, is referred to as a datagram.

In the virtual-circuit approach, a preplanned route is established before any packets are sent. For example, suppose that *A* has one or more messages to send to *E*. It first sends a special control packet, referred to as a Call-Request packet, to 4, requesting a logical connection to *E*. Node 4 decides to route the request and all subsequent packets to 5, which decides to route the request and all subsequent packets to 6, which finally delivers the Call-Request packet to *E*. If *E* is prepared to accept the connection, it sends a Call-Accept packet to 6. This packet is passed back through nodes 5 and 4 to *A*. Stations *A* and *E* may now exchange data over the route that has been established. Because the route is fixed for the duration of the logical connection, it is somewhat similar to a circuit in a circuit-switching network, and is referred to as a virtual circuit. Each packet now contains a virtual-circuit identifier as well as data. Each node on the preestablished route knows where to

direct such packets; no routing decisions are required. Thus, every data packet from *A* intended for E traverses nodes 4, 5, and 6; every data packet from *E* intended for *A* traverses nodes 6, 5, and 4. Eventually, one of the stations terminates the connection with a Clear-Request packet. At any time, each station can have more than one virtual circuit to any other station and can have virtual circuits to more than one station.

So, the main characteristic of the virtual-circuit technique is that a route between stations is set up prior to data transfer. Note that this does not mean that this is a dedicated path, as in circuit switching. A packet is still buffered at each node, and queued for output over a line. The difference from the datagram approach is that, with virtual circuits, the node need not make a routing decision for each packet; it is made only once for all packets using that virtual circuit.

If two stations wish to exchange data over an extended period of time, there are certain advantages to virtual circuits. First, the network may provide services related to the virtual circuit, including sequencing and error control. Sequencing refers to the fact that, because all packets follow the same route, they arrive in the original order. Error control is a service that assures not only that packets arrive in proper sequence, but that all packets arrive correctly. For example, if a packet in a sequence from node 4 to node 6 fails to arrive at node 6, or arrives with an error, node 6 can request a retransmission of that packet from node 4. Another advantage is that packets should transit the network more rapidly with a virtual circuit; it is not necessary to make a routing decision for each packet at each node.

One advantage of the datagram approach is that the call setup phase is avoided. Thus, if a station wishes to send only one or a few packets, datagram delivery will be quicker. Another advantage of the datagram service is that, because it is more primitive, it is more flexible. For example, if congestion develops in one part of the network, incoming datagrams can be routed away from the congestion. With the use of virtual circuits, packets follow a predefined route, and it is thus more difficult for the network to adapt to congestion. A third advantage is that datagram delivery is inherently more reliable. With the use of virtual circuits, if a node fails, all virtual circuits that pass through that node are lost. With datagram delivery, if a node fails, subsequent packets may find an alternate route that bypasses that node.

Most currently available packet-switching networks make use of virtual circuits for their internal operation. To some degree, this reflects a historical motivation to provide a network that presents a service as reliable (in terms of sequencing) as a circuit-switching network. There are, however, several providers of private packet-switching networks that make use of datagram operation. From the user's point of view, there should be very little difference in the external behavior based on the use of datagrams or virtual circuits. If a manager is faced with a choice, other factors such as cost and performance should probably take precedence over whether the internal network operation is datagram or virtual-circuit. Finally, it should be noted that a datagram-style of operation is common in internetworks (discussed in Part IV).

Packet Size

One important design issue is the packet size to be used in the network. There is a significant relationship between packet size and transmission time, as illustrated in

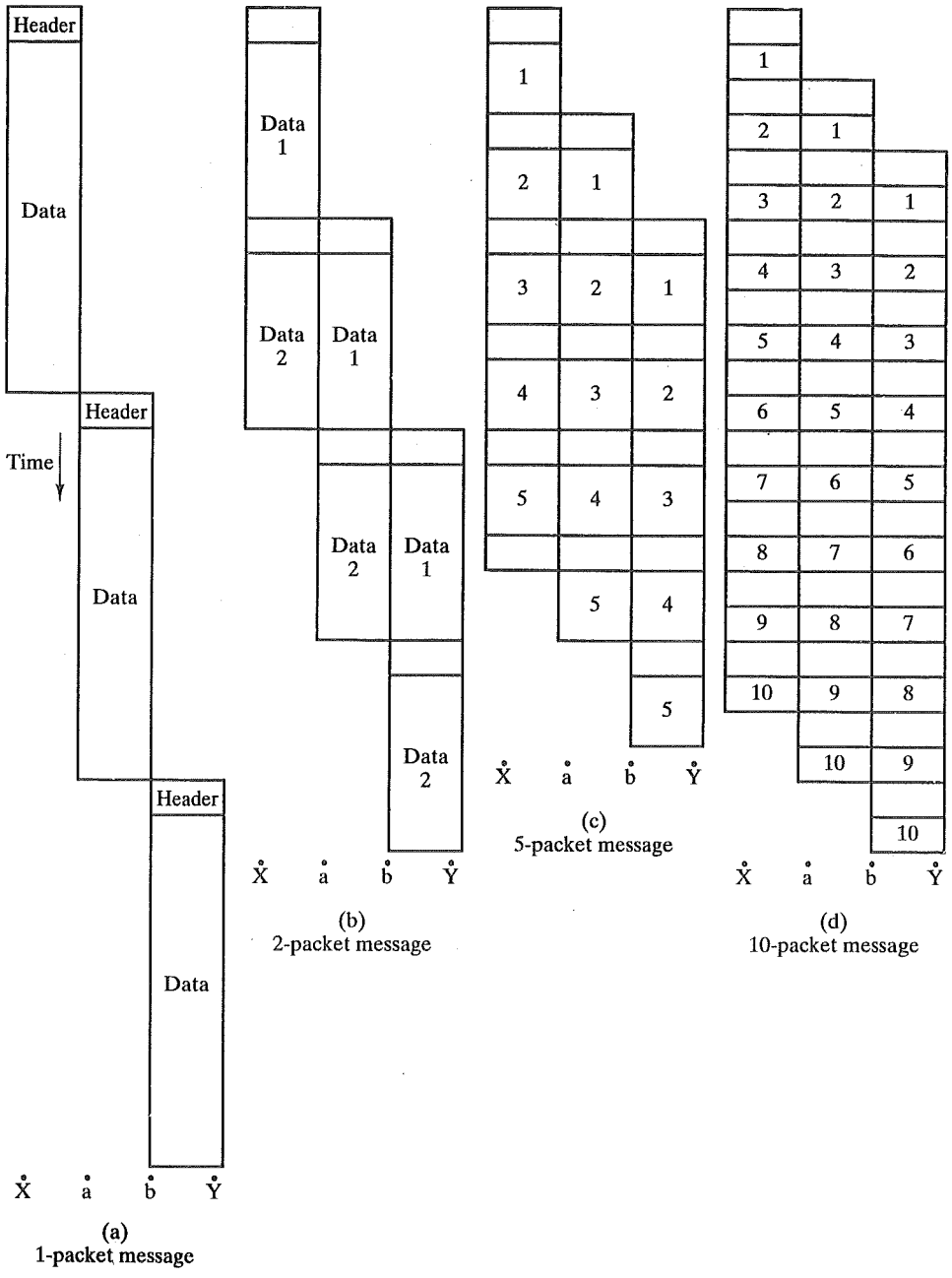


FIGURE 9.2 Effect of packet size on transmission time.

Figure 9.2. In this example, it is assumed that there is a virtual circuit from station *X* through nodes *a* and *b* to station *Y*. The message to be sent comprises 30 octets, and each packet contains 3 octets of control information, which is placed at the

beginning of each packet and is referred to as a *header*. If the entire message is sent as a single packet of 33 octets (3 octets of header plus 30 octets of data), then the packet is first transmitted from station X to node a (Figure 9.2a). When the entire packet is received, it can then be transmitted from a to b. When the entire packet is received at node b, it is then transferred to station Y. The total transmission time at the nodes is 99 octet-times (33 octets $\times$ 3 packet transmissions).

Suppose now that we break up the message into two packets, each containing 15 octets of the message and, of course, 3 octets each of header or control information. In this case, node a can begin transmitting the first packet as soon as it has arrived from X, without waiting for the second packet. Because of this overlap in transmission, the total transmission time drops to 72 octet-times. By breaking the message up into 5 packets, each intermediate node can begin transmission even sooner and the savings in time is greater, with a total of 63 octet-times. However, this process of using more and smaller packets eventually results in increased, rather than reduced, delay as illustrated in Figure 9.2d; this is because each packet contains a fixed amount of header, and more packets means more of these headers. Furthermore, the example does not show the processing and queuing delays at each node. These delays are also greater when more packets are handled for a single message. However, we will see in Chapter 11 that an extremely small packet size (53 octets) can result in an efficient network design.

Comparison of Circuit Switching and Packet Switching

Having looked at the internal operation of packet switching, we can now return to a comparison of this technique with circuit switching. We first look at the important issue of performance, and then examine other characteristics.

Performance

A simple comparison of circuit switching and the two forms of packet switching are provided in Figure 9.3. The figure depicts the transmission of a message across four nodes, from a source station attached to node 1 to a destination station attached to node 4. In this figure, we are concerned with three types of delay:

- **Propagation delay.** The time it takes a signal to propagate from one node to the next. This time is generally negligible. The speed of electromagnetic signals through a wire medium, for example, is typically 2×10^8 m/s.
- **Transmission time.** The time it takes for a transmitter to send out a block of data. For example, it takes 1 s to transmit a 10,000-bit block of data onto a 10-kbps line.
- **Node delay.** The time it takes for a node to perform the necessary processing as it switches data.

For circuit switching, there is a certain amount of delay before the message can be sent. First, a call request signal is sent through the network in order to set up a connection to the destination. If the destination station is not busy, a call-accepted signal returns. Note that a processing delay is incurred at each node during the call request; this time is spent at each node setting up the route of the connection. On

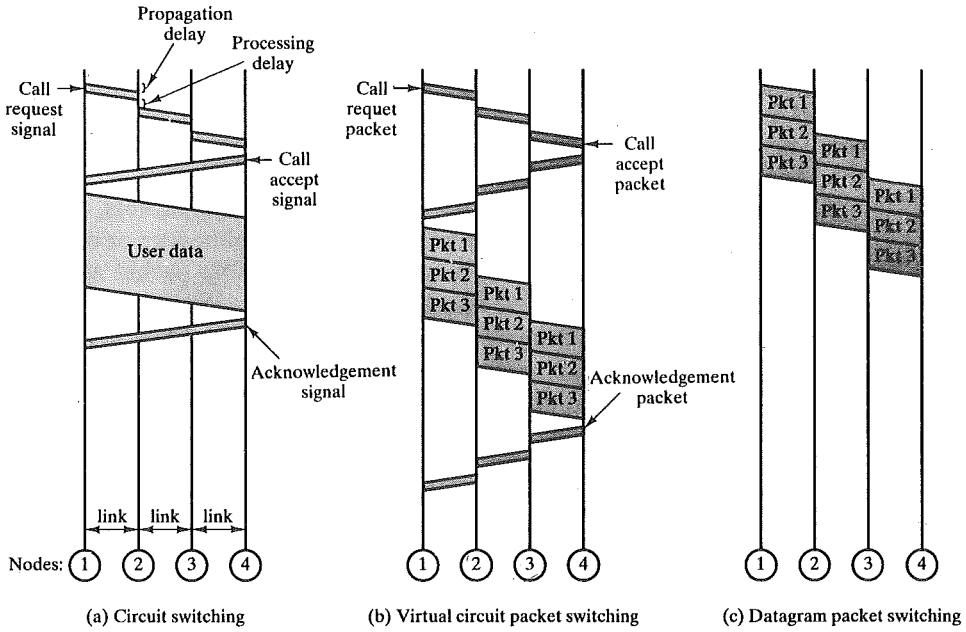


FIGURE 9.3 Event timing for circuit switching and packet switching.

the return, this processing is not needed because the connection is already set up; once it is set up, the message is sent as a single block, with no noticeable delay at the switching nodes.

Virtual-circuit packet switching appears quite similar to circuit switching. A virtual circuit is requested using a call-request packet, which incurs a delay at each node. The virtual circuit is accepted with a call-accept packet. In contrast to the circuit-switching case, the call acceptance also experiences node delays, even though the virtual circuit route is now established; the reason is that this packet is queued at each node and must wait its turn for retransmission. Once the virtual circuit is established, the message is transmitted in packets. It should be clear that this phase of the operation can be no faster than circuit switching, for comparable networks; this is because circuit switching is an essentially transparent process, providing a constant data rate across the network. Packet switching involves some delay at each node in the path; worse, this delay is variable and will increase with increased load.

Datagram packet switching does not require a call setup. Thus, for short messages, it will be faster than virtual-circuit packet switching and perhaps circuit switching. However, because each individual datagram is routed independently, the processing for each datagram at each node may be longer than for virtual-circuit packets. Thus, for long messages, the virtual-circuit technique may be superior.

Figure 9.3 is intended only to suggest what the relative performance of the techniques might be; however, actual performance depends on a host of factors, including the size of the network, its topology, the pattern of load, and the characteristics of typical exchanges.

TABLE 9.1 Comparison of communication switching techniques.

Circuit switching	Datagram packet switching	Virtual-circuit packet switching
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth transmission	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each message	Overhead bits in each packet

Other Characteristics

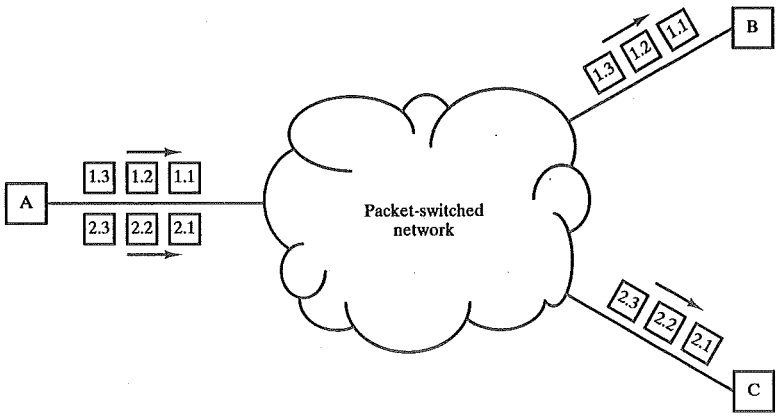
Besides performance, there are a number of other characteristics that may be considered in comparing the techniques we have been discussing. Table 9.1 summarizes the most important of these. Most of these characteristics have already been discussed. A few additional comments follow.

As was mentioned, circuit switching is essentially a transparent service. Once a connection is established, a constant data rate is provided to the connected stations; this is not the case with packet switching, which typically introduces variable delay, so that data arrive in a choppy manner. Indeed, with datagram packet switching, data may arrive in a different order than they were transmitted.

An additional consequence of transparency is that there is no overhead required to accommodate circuit switching. Once a connection is established, the analog or digital data are passed through, as is, from source to destination. For packet switching, analog data must be converted to digital before transmission; in addition, each packet includes overhead bits, such as the destination address.

External and Internal Operation

One of the most important characteristics of a packet-switching network is whether it uses datagrams or virtual circuits. Actually, there are two dimensions of this characteristic, as illustrated in Figure 9.4. At the interface between a station and a network node, a network may provide either a connection-oriented or connectionless service. With a connection-oriented service, a station performs a call request to set up a logical connection to another station. All packets presented to the network are identified as belonging to a particular logical connection and are numbered sequentially. The network undertakes to deliver packets in sequence-number order. The logical connection is usually referred to as a virtual circuit, and the connection-oriented service is referred to as an *external virtual-circuit service*; unfortunately, this external service is distinct from the concept of *internal virtual-circuit operation*, as we shall see. An important example of an external virtual circuit service is X.25, which is examined in Section 9.4.



(a) External virtual circuit. A logical connection is set up between two stations. Packets are labeled with a virtual circuit number and a sequence number. Packets arrive in sequence.

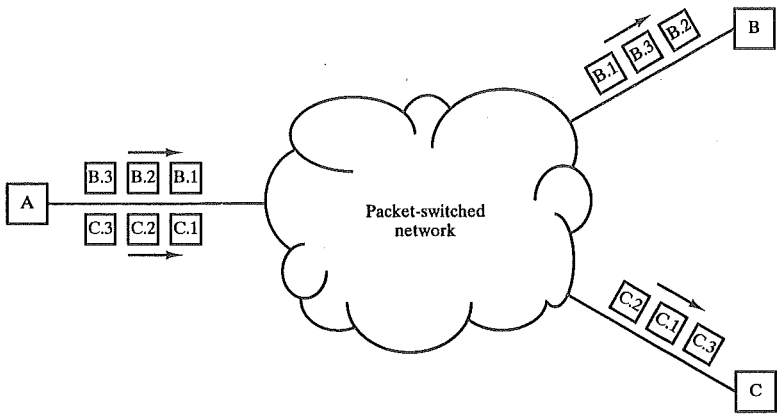
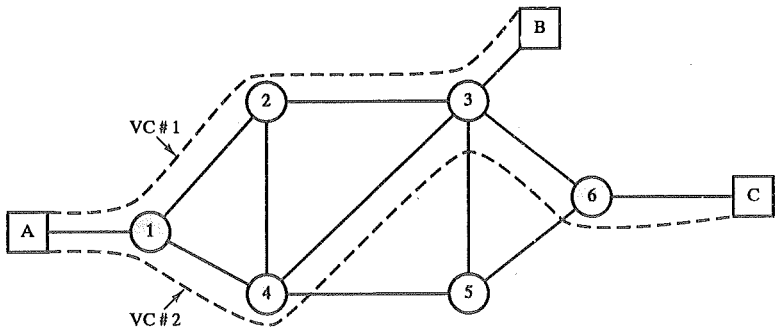


FIGURE 9.4 External and internal virtual circuits and datagrams.
(*continued on next page*)

With connectionless service, the network only agrees to handle packets independently, and may not deliver them in order or reliably. This type of service is sometimes known as an *external datagram service*; again, this concept is distinct from that of *internal datagram operation*. Internally, the network may actually construct a fixed route between endpoints (virtual circuit), or it may not (datagram).

These internal and external design decisions need not coincide:

- **External virtual circuit, internal virtual circuit.** When the user requests a virtual circuit, a dedicated route through the network is constructed. All packets follow that same route.
- **External virtual circuit, internal datagram.** The network handles each packet separately. Thus, different packets for the same external virtual circuit may take different routes. However, the network buffers packets at the destination node, if necessary, so that they are delivered to the destination station in the proper order.
- **External datagram, internal datagram.** Each packet is treated independently from both the user's and the network's point of view.
- **External datagram, internal virtual circuit.** The external user does not see any connections, as it simply sends packets one at a time. The network, however, sets up a logical connection between stations for packet delivery and may



(c) Internal virtual circuit. A route for packets between two stations is defined and labeled. All packets for that virtual circuit follow the same route and arrive in sequence.

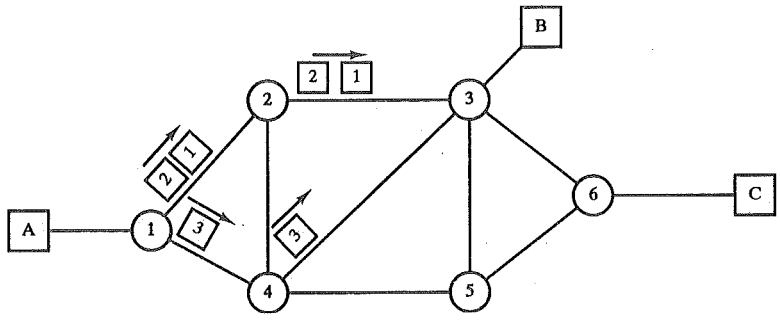


FIGURE 9.4 *(continued)*

leave such connections in place for an extended period, so as to satisfy anticipated future needs.

The question arises as to the choice of virtual circuits or datagrams, both internally and externally. This will depend on the specific design objectives for the communication network and the cost factors that prevail.

We have already made some comments concerning the relative merits of internal datagram versus virtual-circuit operation. With respect to external service, we can make the following observations.

- The datagram service, coupled with internal datagram operation, allows for efficient use of the network; no call setup and no need to hold up packets while a packet in error is retransmitted. This latter feature is desirable in some real-time applications.
- The virtual-circuit service can provide end-to-end sequencing and error control; this service is attractive for supporting connection-oriented applications, such as file transfer and remote-terminal access.

In practice, the virtual-circuit service is much more common than the datagram service. The reliability and convenience of a connection-oriented service is seen as more attractive than the benefits of the datagram service.

9.2 ROUTING

One of the most complex and crucial aspects of packet-switching network design is routing. This section begins with a survey of key characteristics that can be used to classify routing strategies. Then, some specific routing strategies are discussed.

The principles described in this section are also applicable to internetwork routing, discussed in Part III.

Characteristics

The primary function of a packet-switching network is to accept packets from a source station and deliver them to a destination station. To accomplish this, a path or route through the network must be determined; generally, more than one route is possible. Thus, a routing function must be performed. The requirements for this function include

- Correctness
- Simplicity
- Robustness
- Stability
- Fairness
- Optimality
- Efficiency

The first two items on the list are self-explanatory. Robustness has to do with the ability of the network to deliver packets via some route in the face of localized

failures and overloads. Ideally, the network can react to such contingencies without the loss of packets or the breaking of virtual circuits. The designer who seeks robustness must cope with the competing requirement for stability. Techniques that react to changing conditions have an unfortunate tendency to either react too slowly to events or to experience unstable swings from one extreme to another. For example, the network may react to congestion in one area by shifting most of the load to a second area. Now the second area is overloaded and the first is underutilized, causing a second shift. During these shifts, packets may travel in loops through the network.

A tradeoff also exists between fairness and optimality. Some performance criteria may give higher priority to the exchange of packets between nearby stations compared to an exchange between distant stations. This policy may maximize average throughput but will appear unfair to the station that primarily needs to communicate with distant stations.

Finally, any routing technique involves some processing overhead at each node and often a transmission overhead as well, both of which impair network efficiency. The penalty of such overhead needs to be less than the benefit accrued based on some reasonable metric, such as increased robustness or fairness.

With these requirements in mind, we are in a position to assess the various design elements that contribute to a routing strategy. Table 9.2 lists these elements. Some of these categories overlap or are dependent on one another. Nevertheless, an examination of this list serves to clarify and organize routing concepts.

TABLE 9.2 Elements of routing techniques for packet-switching networks.

Performance criteria	Network information source
Number of hops	None
Cost	Local
Delay	Adjacent node
Throughput	Nodes along route
	All nodes
Decision time	
Packet (datagram)	Network information update timing
Session (virtual circuit)	Continuous
	Periodic
Decision place	Major load change
Each node (distributed)	Topology change
Central node (centralized)	
Originating node (source)	

Performance Criteria

The selection of a route is generally based on some performance criterion. The simplest criterion is to choose the minimum-hop route (one that passes through the least number of nodes) through the network; this is an easily measured criterion and should minimize the consumption of network resources. A generalization of the minimum-hop criterion is least-cost routing. In this case, a cost is associated with each link, and, for any pair of attached stations, the route through the network that accumulates the least cost is sought. For example, Figure 9.5 illustrates a net-

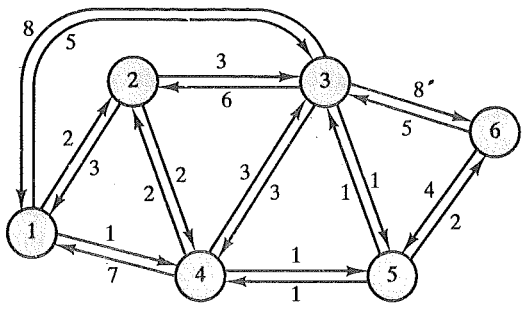


FIGURE 9.5 Example packet-switched network.

work in which the two arrowed lines between a pair of nodes represent a link between this nodes, and the corresponding numbers represent the current link cost in each direction. The shortest path (fewest hops) from node 1 to node 6 is 1-3-6 (cost = 5 + 5 = 10), but the least-cost path is 1-4-5-6 (cost = 1 + 1 + 2 = 4). Costs are assigned to links to support one or more design objectives. For example, the cost could be inversely related to the data rate (i.e., the higher the data rate on a link, the lower the assigned cost of the link) or the current queuing delay on the link. In the first case, the least-cost route should provide the highest throughput. In the second case, the least-cost route should minimize delay.

In either the minimum-hop or least-cost approach, the algorithm for determining the optimum route for any pair of stations is relatively straightforward, and the processing time would be about the same for either computation. Because the least-cost criterion is more flexible, it is more common than the minimum-hop criterion.

Several least-cost routing algorithms are in common use. These are described in Appendix 9A.

Decision Time and Place

Routing decisions are made on the basis of some performance criterion. Two key characteristics of the decision are the time and place that the decision is made.

Decision time is determined by whether the routing decision is made on a packet or virtual-circuit basis. When the internal operation of the network is datagram, a routing decision is made individually for each packet. For internal virtual-circuit operation, a routing decision is made at the time the virtual circuit is established. In the simplest case, all subsequent packets using that virtual circuit follow the same route. In more sophisticated network designs, the network may dynamically change the route assigned to a particular virtual circuit in response to changing conditions (e.g., overload or failure of a portion of the network).

The term *decision place* refers to which node or nodes in the network are responsible for the routing decision. Most common is distributed routing, in which each node has the responsibility of selecting an output link for routing packets as they arrive. For centralized routing, the decision is made by some designated node, such as a network control center. The danger of this latter approach is that the loss

of the network control center may block operation of the network. The distributed approach is perhaps more complex, but is also more robust. A third alternative, used in some networks, is source routing. In this case, the routing decision is actually made by the source station rather than by a network node, and is then communicated to the network; this allows the user to dictate a route through the network that meets criteria local to that user.

The decision time and decision place are independent design variables. For example, in Figure 9.5, suppose that the decision place is each node and that the values depicted are the costs at a given instant in time; the costs, though, may change. If a packet is to be delivered from node 1 to node 6, it might follow the route 1-4-5-6, with each leg of the route determined locally by the transmitting node. Now let the values change such that 1-4-5-6 is no longer the optimum route. In a datagram network, the next packet may follow a different route, again determined by each node along the way, In a virtual-circuit network, each node will remember the routing decision that was made when the virtual circuit was established, and will simply pass on the packets without making a new decision.

Network Information Source and Update Timing

Most routing strategies require that decisions be based on knowledge of the topology of the network, traffic load, and link cost. Surprisingly, some strategies use no such information and yet manage to get packets through; flooding and some random strategies (discussed below) are in this category.

With distributed routing, in which the routing decision is made by each node, the individual node may make use of only local information, such as the cost of each outgoing link. Each node might also collect information from adjacent (directly connected) nodes, such as the amount of congestion experienced at that node. Finally, there are algorithms in common use that allow the node to gain information from all nodes on any potential route of interest. In the case of centralized routing, the central node typically makes use of information obtained from all nodes.

A related concept is that of information update timing, which is a function of both the information source and the routing strategy. Clearly, if no information is used (as in flooding), there is no information to update. If only local information is used, the update is essentially continuous—that is, an individual node always knows its local conditions. For all other information source categories (adjacent nodes, all nodes), update timing depends on the routing strategy. For a fixed strategy, the information is never updated. For an adaptive strategy, information is updated from time to time to enable the routing decision to adapt to changing conditions.

As you might expect, the more information available, and the more frequently it is updated, the more likely the network is to make good routing decisions. On the other hand, the transmission of that information consumes network resources.

Routing Strategies

A large number of routing strategies have evolved for dealing with the routing requirements of packet-switching networks; many having these strategies are also

applied to internetwork routing, which we cover in Part III. In this section, we survey four key strategies: fixed, flooding, random, and adaptive.

Fixed Routing

For fixed routing, a route is selected for each source-destination pair of nodes in the network. Either of the least-cost routing algorithms described in Appendix 9A could be used. The routes are fixed, with the exception that they might change if there is movement in the topology of the network. Thus, the link costs used in designing routes cannot be based on any dynamic variable such as traffic. They could, however, be based on expected traffic or capacity.

Figure 9.6 suggests how fixed routing might be implemented. A central routing matrix is created, to be stored perhaps at a network control center. The matrix shows, for each source-destination pair of nodes, the identity of the next node on the route.

Note that it is not necessary to store the complete route for each possible pair of nodes. Rather, it is sufficient to know, for each pair of nodes, the identity of the first node on the route; to see this, suppose that the least-cost route from X to Y begins with the X-A link. Call the remainder of the route R_1; this is the part from A

CENTRAL ROUTING DIRECTORY

To Node \ From Node	1	2	3	4	5	6
1	—	1	5	1	4	5
2	2	—	2	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
2	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
2	5
3	2
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
2	1
3	2
4	5
5	5
6	5

Node 5 Directory

Destination	Next Node
2	4
3	4
4	3
5	4
6	6

Node 6 Directory

Destination	Next Node
2	5
3	5
4	5
5	5
6	5

FIGURE 9.6 Fixed routing (using Figure 9.5).

to Y. Define R_2 as the least-cost route from A to Y. Now, if the cost of R_1 is greater than that of R_2, then the X-Y route can be improved by using R_2 instead. If the cost of R_1 is less than R_2, then R_2 is not the least-cost route from A to Y. Therefore, $R_1 = R_2$. Thus, at each point along a route, it is only necessary to know the identity of the next node, not the entire route. In our example, the route from node 1 to node 6 begins by going through node 4. Again, consulting the matrix, the route from node 4 to node 6 goes through node 5. Finally, the route from node 5 to node 6 is a direct link to node 6. The complete route, then, from node 1 to node 6 is 1-4-5-6.

From this overall matrix, routing tables can be developed and stored at each node. From the reasoning in the preceding paragraph, it follows that each node need only store a single column of the routing directory. The node's directory shows the next node to take for each destination.

With fixed routing, there is no difference between routing for datagrams and virtual circuits. All packets from a given source to a given destination follow the same route. The advantage of fixed routing is its simplicity, and it should work well in a reliable network with a stable load. Its disadvantage is its lack of flexibility; it does not react to network congestion or failures.

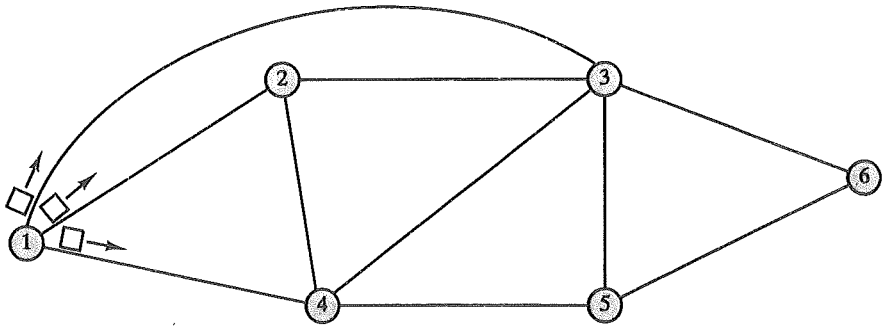
A refinement to fixed routing that would accommodate link and node outages would be to supply the nodes with an alternate next node for each destination. For example, the alternate next nodes in the node 1 directory might be 4, 3, 2, 3, 3.

Flooding

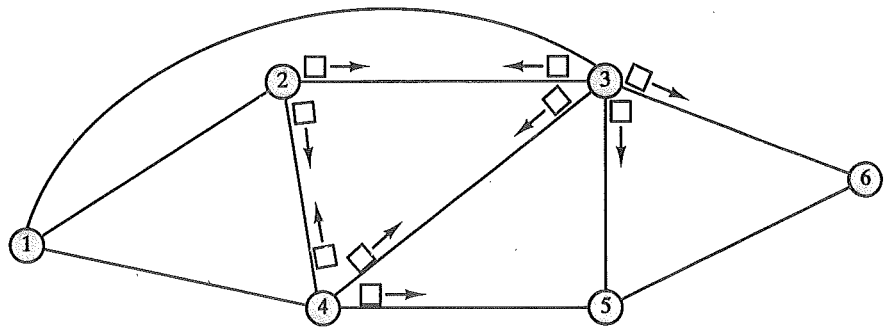
Another simple routing technique is flooding. This technique requires no network information whatsoever, and works as follows. A packet is sent by a source node to every one of its neighbors. At each node, an incoming packet is retransmitted on all outgoing links except for the link on which it arrived. For example, if node 1 in Figure 9.5 has a packet to send to node 6, it sends a copy of that packet (with a destination address of 6), to nodes 2, 3, and 4. Node 2 will send a copy to nodes 3 and 4. Node 4 will send a copy to nodes 2, 3, and 5. And so it goes. Eventually, a number of copies of the packet will arrive at node 6. The packet must have some unique identifier (e.g., source node and sequence number, or virtual-circuit number and sequence number) so that node 6 knows to discard all but the first copy.

Unless something is done to stop the incessant retransmission of packets, the number of packets in circulation just from a single source packet grows without bound; one way to prevent this is for each node to remember the identity of those packets it has already retransmitted. When duplicate copies of the packet arrive, they are discarded. A simpler technique is to include a hop count field with each packet. The count can originally be set to some maximum value, such as the diameter (length of the longest minimum-hop path through the network) of the network. Each time a node passes on a packet, it decrements the count by one. When the count reaches zero, the packet is discarded.

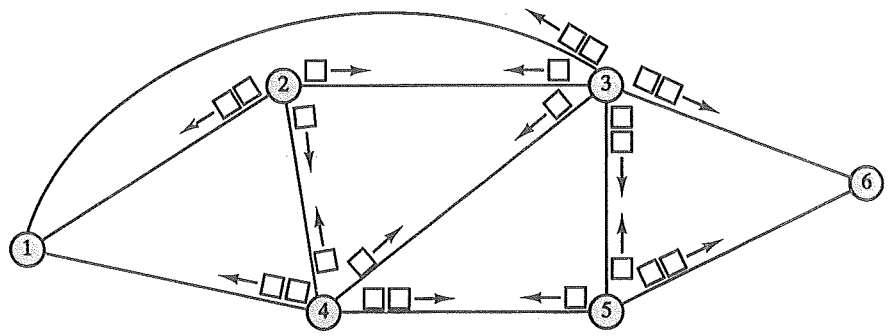
An example of the latter tactic is shown in Figure 9.7. A packet is to be sent from node 1 to node 6 and is assigned a hop count of 3. On the first hop, three copies of the packet are created. For the second hop of all these copies, a total of nine copies are created. One of these copies reaches node 6, which recognizes that it is the intended destination and does not retransmit. However, the other nodes generate a total of 22 new copies for their third and final hop. Note that if a node is not



(a) First hop



(b) Second hop



(c) Third hop

FIGURE 9.7 Flooding example (hop count = 3).

keeping track of the packet identifier, it may generate multiple copies at this third stage. All packets received from the third hop are discarded. In all, node 6 has received four additional copies of the packet.

The flooding technique has three remarkable properties:

- All possible routes between source and destination are tried. Thus, no matter what link or node outages have occurred, a packet will always get through if at least one path between source and destination exists.

- Because all routes are tried, at least one copy of the packet to arrive at the destination will have used a minimum-hop route.
- All nodes that are directly or indirectly connected to the source node are visited.

Because of the first property, the flooding technique is highly robust and could be used to send emergency messages. An example application is a military network that is subject to extensive damage. Because of the second property, flooding might be used to initially set up the route for a virtual circuit. The third property suggests that flooding can be useful for the dissemination of important information to all nodes; we will see that it is used in some schemes to disseminate routing information.

The principal disadvantage of flooding is the high traffic load that it generates, which is directly proportional to the connectivity of the network.

Random Routing

Random routing has the simplicity and robustness of flooding with far less traffic load. With random routing, a node selects only one outgoing path for retransmission of an incoming packet. The outgoing link is chosen at random, excluding the link on which the packet arrived. If all links are equally likely to be chosen, then a node may simply utilize outgoing links in a round-robin fashion.

A refinement of this technique is to assign a probability to each outgoing link and to select the link based on that probability. The probability could be based on data rate, in which case we have

$$P_i = \frac{R_i}{\Sigma_j R_j}$$

where

P_i = probability of selecting link i
R_i = data rate on link i

The sum is taken over all candidate outgoing links. This scheme should provide good traffic distribution. Note that the probabilities could also be based on fixed link costs.

Like flooding, random routing requires the use of no network information. Because the route taken is random, the actual route will typically not be the least-cost route nor the minimum-hop route. Thus, the network must carry a higher than optimum traffic load, although not nearly as high as for flooding.

Adaptive Routing

In virtually all packet-switching networks, some sort of adaptive routing technique is used. That is, the routing decisions that are made change as conditions on the network change. The principle conditions that influence routing decisions are

- **Failure.** When a node or trunk fails, it can no longer be used as part of a route.
- **Congestion.** When a particular portion of the network is heavily congested,

it is desirable to route packets around, rather than through, the area of congestion.

For adaptive routing to be possible, information about the state of the network must be exchanged among the nodes. There is a tradeoff here between the quality of the information and the amount of overhead. The more information that is exchanged, and the more frequently it is exchanged, the better will be the routing decisions that each node makes. On the other hand, this information is itself a load on the network, causing a performance degradation.

There are several drawbacks associated with the use of adaptive routing:

- The routing decision is more complex; therefore, the processing burden on network nodes increases.
- In most cases, adaptive strategies depend on status information that is collected at one place but used at another; therefore, the traffic burden on the network increases.
- An adaptive strategy may react too quickly, causing congestion-producing oscillation; if it reacts too slowly, the strategy will be irrelevant.

Despite these real dangers, adaptive routing strategies are by far the most prevalent, for two reasons:

- An adaptive routing strategy can improve performance, as seen by the network user.
- An adaptive routing strategy can aid in congestion control, as discussed later.

These benefits may or may not be realized, depending on the soundness of the design and the nature of the load. By and large, it is an extraordinarily complex task to perform properly. As demonstration of this, most major packet-switching networks, such as ARPANET and its successors, TYMNET, and those developed by IBM and DEC, have endured at least one major overhaul of their routing strategy.

A convenient way to classify adaptive routing strategies is on the basis of information source: local, adjacent nodes, all nodes. An example of an adaptive routing strategy that relies only on local information is one in which a node routes each packet to the outgoing link with the shortest queue length, Q. This would have the effect of balancing the load on outgoing links. However, some outgoing links may not be headed in the correct general direction. We can improve matters by also taking into account preferred direction, much as with random routing. In this case, each link emanating from the node would have a bias B_i, for each destination i. For each incoming packet headed for node i, the node would choose the outgoing link that minimizes $Q + B_i$. Thus, a node would tend to send packets in the right direction, with a concession made to current traffic delays.

As an example, Figure 9.8 shows the status of node 4 of Figure 9.5 at a certain point in time. Node 4 has links to four other nodes. A fair number of packets have been arriving and a backlog has built up, with a queue of packets waiting for each of the outgoing links. A packet arrives from node 1 destined for node 6. To which outgoing link should the packet be routed? Based on current queue lengths and the

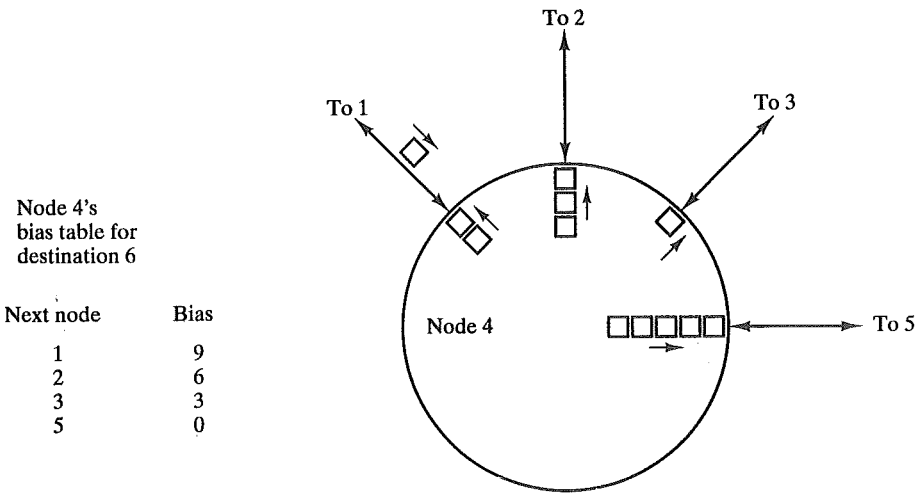


FIGURE 9.8 Example of isolated adaptive routing.

values of bias (B_6) for each outgoing link, the minimum value of $Q + B_6$ is 4, on the link to node 3. Thus, node 4 routes the packet through node 3.

Adaptive schemes based only on local information are rarely used because they do not exploit easily available information. Strategies based on information from adjacent nodes or all nodes are commonly found. Both take advantage of information that each node has about delays and outages that it experiences. Such adaptive strategies can be either distributed or centralized. In the distributed case, each node exchanges delay information with other nodes. Based on incoming information, a node tries to estimate the delay situation throughout the network, and applies a least-cost routing algorithm. In the centralized case, each node reports its link delay status to a central node, which designs routes based on this incoming information and sends the routing information back to the nodes.

Examples

In this section, we look at several examples of routing strategies. All of these were initially developed for ARPANET, which is a packet-switching network that was the foundation of the present-day Internet. It is instructive to examine these strategies for several reasons. First, these strategies, and similar ones, are also used in other packet-switching networks, including those developed by DEC and IBM and including a number of networks on the Internet. Second, routing schemes based on the ARPANET work have also been used for internetwork routing the Internet and in private internetworks. And finally, the ARPANET routing scheme evolved in a way that illuminates some of the key design issues related to routing algorithms.

First Generation

The original routing algorithm, designed in 1969, was a distributed adaptive algorithm using estimated delay as the performance criterion and a version of the

Bellman-Ford algorithm (Appendix 9A). For this algorithm, each node maintains two vectors:

$$D_i = \begin{bmatrix} d_{i1} \\ \cdot \\ \cdot \\ \cdot \\ d_{iN} \end{bmatrix} \qquad S_i = \begin{bmatrix} s_{i1} \\ \cdot \\ \cdot \\ \cdot \\ s_{iN} \end{bmatrix}$$

where

D_i = delay vector for node i
d_{ij} = current estimate of minimum delay from node i to node j ($d_{ii} = 0$)
N = number of nodes in the network
S_i = successor node vector for node i
s_{ij} = the next node in the current minimum-delay route from i to j

Periodically (every 128 ms), each node exchanges its delay vector with all of its neighbors. On the basis of all incoming delay vectors, a node k updates both of its vectors as follows:

$$d_{kj} = \min_{i \in A}[d_{ij} + l_{ki}]$$
$$s_{kj} = i \qquad \text{using } i \text{ that minimizes the expression above}$$

where

A = set of neighbor nodes for k
l_{ki} = current estimate of delay from k to i

Figure 9.9 provides an example of the original ARPANET algorithm, using the network of Figure 9.10. This is the same network as that of Figure 9.5, with some of the link costs having different values (and assuming the same cost in both directions). Figure 9.9a shows the routing table for node 1 at an instant in time that reflects the link costs of Figure 9.10. For each destination, a delay is specified, as well as the next node on the route that produces that delay. At some point, the link costs change to those of Figure 9.5. Assume that node 1's neighbors (nodes 2, 3, and 4) learn of the change before node 1. Each of these nodes updates its delay vector and sends a copy to all of its neighbors, including node 1 (Figure 9.9b). Node 1 discards its current routing table and builds a new one, based solely on the incoming delay vector and its own estimate of link delay to each of its neighbors. The result is shown in Figure 9.9c.

The estimated link delay is simply the queue length for that link. Thus, in building a new routing table, the node will tend to favor outgoing links with shorter queues. This tends to balance the load on outgoing links. However, because queue lengths vary rapidly with time, the distributed perception of the shortest route could

Destination	Delay	Next node
1	0	–
2	2	2
3	5	3
4	1	4
5	6	3
6	8	3

D_1 S_1

(a) Node 1's routing table before update

D_2	D_3	D_4
2	3	1
0	3	2
3	0	2
2	2	0
3	1	1
5	3	3

(b) Delay vectors sent to node 1 from neighbor nodes

Destination	Delay	Next node
1	0	–
2	2	2
3	3	4
4	1	4
5	2	4
6	4	4

$l_{1,2} = 2$
$l_{1,3} = 5$
$l_{1,4} = 1$

(c) Node 1's routing table after update and link costs used in update

FIGURE 9.9 Original ARPANET routing algorithm.

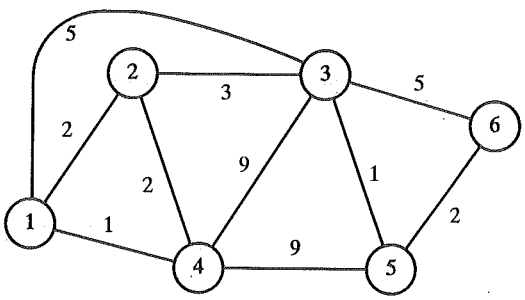


FIGURE 9.10 Network for example of Figure 9.9a.

change while a packet is en route; this could lead to a thrashing situation in which a packet continues to seek out areas of low congestion rather than aiming at the destination.

Second Generation

After some years of experience and several minor modifications, the original routing algorithm was replaced by a quite different one in 1979 [MCQU80]. The major shortcomings of the old algorithm were these:

- The algorithm did not consider line speed, but merely queue length. Thus, higher-capacity links were not given the favored status they deserved.
- Queue length is, in any case, an artificial measure of delay, as some variable amount of processing time elapses between the arrival of a packet at a node and its placement in an outbound queue.
- The algorithm was not very accurate. In particular, it responded slowly to congestion and delay increases.

The new algorithm is also a distributed adaptive one, using delay as the performance criterion, but the differences are significant. Rather than using queue length as a surrogate for delay, the delay is measured directly. At a node, each incoming packet is timestamped with an arrival time. A departure time is recorded when the packet is transmitted. If a positive acknowledgment is returned, the delay for that packet is recorded as the departure time minus the arrival time plus transmission time and propagation delay. The node must therefore know link data rate and propagation time. If a negative acknowledgment comes back, the departure time is updated and the node tries again, until a measure of successful transmission delay is obtained.

Every 10 seconds, the node computes the average delay on each outgoing link. If there are any significant changes in delay, the information is sent to all other nodes using flooding. Each node maintains an estimate of delay on every network link. When new information arrives, it recomputes its routing table using Dijkstra's algorithm (Appendix 9A).

Third Generation

Experience with this new strategy indicated that it was more responsive and stable than the old one. The overhead induced by flooding was moderate as each node does this, at most, once every 10 seconds. However, as the load on the network grew, a shortcoming in the new strategy began to appear, and the strategy was revised in 1987 [KHAN89].

The problem with the second strategy is the assumption that the measured packet delay on a link is a good predictor of the link delay encountered after all nodes reroute their traffic based on this reported delay. Thus, it is an effective routing mechanism only if there is some correlation between the reported values and those actually experienced after rerouting. This correlation tends to be rather high under light and moderate traffic loads. However, under heavy loads, there is little correlation. Therefore, immediately after all nodes have made routing updates, the routing tables are obsolete!

As an example, consider a network that consists of two regions with only two links, A and B, connecting the two regions (Figure 9.11). Each route between two nodes in different regions must pass through one of these links. Assume that a situation develops in which most of the traffic is on link A. This will cause the link delay on A to be significant, and, at the next opportunity, this delay value will be reported to all other nodes. These updates will arrive at all nodes at about the same time, and all will update their routing tables immediately. It is likely that this new delay value for link A will be high enough to make link B the preferred choice for most, if not

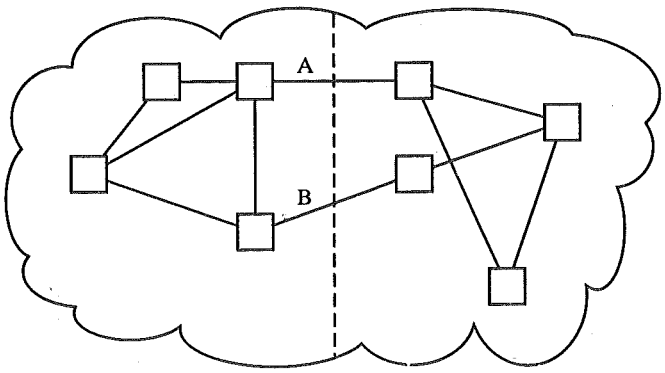


FIGURE 9.11 Packet-switching network subject to oscillations.

all, interregion routes. Because all nodes adjust their routes at the same time, most or all interregion traffic shifts at the same time to link B. Now, the link delay value on B will become high, and there will be a subsequent shift to link A. This oscillation will continue until the traffic volume subsides.

There are a number of reasons why this oscillation is undesirable:

1. A significant portion of available capacity is unused at just the time when it is needed most: under heavy traffic load.
2. The overutilization of some links can lead to the spread of congestion within the network. (This will be seen in the discussion of congestion in Section 9.3.)
3. The large swings in measured delay values result in the need for more frequent routing update messages; this increases the load on the network at just the time when the network is already stressed.

The ARPANET designers concluded that the essence of the problem was that every node was trying to obtain the best route for all destinations, and that these efforts conflicted. It was concluded that under heavy loads, the goal of routing should be to give the average route a good path instead of attempting to give all routes the best path.

The designers decided that it was unnecessary to change the overall routing algorithm. Rather, it was sufficient to change the function that calculates link costs. This was done in such a way as to damp routing oscillations and reduce routing overhead. The calculation begins with measuring the average delay over the last 10 seconds. This value is then transformed with the following steps:

1. Using a simple M/M/1 queuing model, the measured delay is transformed into an estimate of link utilization. From queuing theory, utilization can be expressed as a function of delay as follows:

$$\rho = \frac{2(s - t)}{s - 2t}$$

where

ρ = link utilization
t = measured delay
s = service time

The service time was set at the network-wide average packet size (600 bits) divided by the data rate of the link.

2. The result is then smoothed by averaging it with the previous estimate of utilization:

$$U(n + 1) = 0.5 \times \rho(n + 1) + 0.5 \times U(n)$$

where

$U(n)$ = average utilization calculated at sampling time n
$\rho(n)$ = link utilization measured at sampling time n

Averaging increases the period of routing oscillations, thus reducing routing overhead.

3. The link cost is then set as a function of average utilization that is designed to provide a reasonable estimate of cost while avoiding oscillation. Figure 9.12 indicates the way in which the estimate of utilization is converted into a cost value. The final cost value is, in effect, a transformed value of delay.

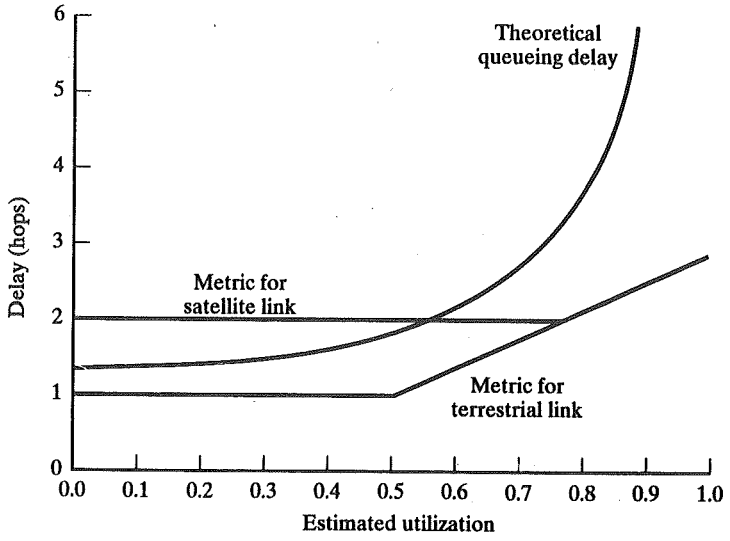


FIGURE 9.12 ARPANET delay metrics.

In the figure, delay is normalized to the value achieved on an idle line, which is just propagation delay plus transmission time. One curve on the figure indicates the way in which the actual delay rises as a function of utilization; the increase in delay is due to queuing delay at the node. For the revised algorithm, the cost value is kept at the minimum value until a given level of utilization is reached. This feature has the effect of reducing routing overhead at low traffic levels. Above a certain level of utilization, the cost level is allowed to rise to a maximum value that is equal to three times the minimum value. The effect of this maximum value is to dictate that traffic should not be routed around a heavily utilized line by more than two additional hops.

Note that the minimum threshold is set higher for satellite links; this encourages the use of terrestrial links under conditions of light traffic, as the terrestrial links have much lower propagation delay. Note also that the actual delay curve is much steeper than the transformation curves at high utilization levels. It is this steep rise in link cost which that all of the traffic on a link to be shed, which in turn causes routing oscillations.

In summary, the revised cost function is keyed to utilization rather than delay. The function resembles a delay-based metric under light loads, as well as a capacity-based metric under heavy loads.

9.3 CONGESTION CONTROL

As with routing, the concept of traffic control in a packet-switching network is complex, and a wide variety of approaches have been proposed. The objective here is to maintain the number of packets within the network below the level at which performance falls off dramatically.

To understand the issue involved in congestion control, we need to look at some results from queuing theory. In essence, a packet-switching network is a network of queues. At each node, there is a queue of packets for each outgoing channel. If the rate at which packets arrive and queue up exceeds the rate at which packets can be transmitted, the queue size grows without bound and the delay experienced by a packet goes to infinity. Even if the packet arrival rate is less than the packet transmission rate, queue length will grow dramatically as the arrival rate approaches the transmission rate. We saw this kind of behavior in Figure 7.16. As a rule of thumb, when the line for which packets are queuing becomes more than 80% utilized, the queue length grows at an alarming rate.

Consider the queuing situation at a single packet-switching node, such as is illustrated in Figure 9.13. Any given node has a number of transmission links attached to it: one or more to other packet-switching nodes, and zero or more to host systems. On each link, packets arrive and depart. We can consider that there are two buffers at each link, one to accept arriving packets, and one to hold packets that are waiting to depart. In practice, there might be two fixed-size buffers associated with each link, or there might be a pool of memory available for all buffering activities. In the latter case, we can think of each link having two variable-size

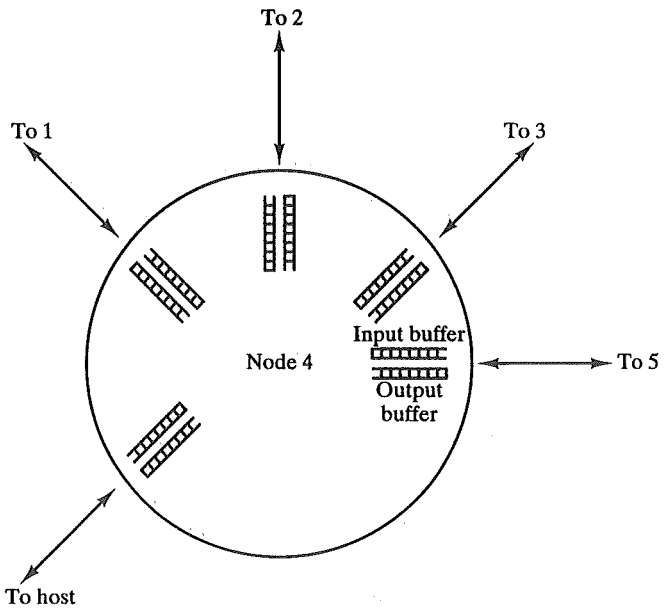


FIGURE 9.13 Input and output queues at node 4 of Figure 9.5.

buffers associated with it, subject to the constraint that the sum of all buffer sizes is a constant.

In any case, as packets arrive, they are stored in the input buffer of the corresponding link. The node examines each incoming packet to make a routing decision, and then moves the packet to the appropriate output buffer. Packets queued up for output are transmitted as rapidly as possible; this is, in effect, statistical time-division multiplexing. Now, if packets arrive too fast for the node to process them (make routing decisions), or faster than packets can be cleared from the outgoing buffers, then, eventually, packets will arrive for which no memory is available.

When such a saturation point is reached, one of two general strategies can be adopted. The first such strategy is to simply discard any incoming packet for which there is no available buffer space. The alternative is for the node that is experiencing these problems to exercise some sort of flow control over its neighbors so that the traffic flow remains manageable. But, as Figure 9.14 illustrates, each of a node's neighbors is also managing a number of queues. If node 6 restrains the flow of packets from node 5, this causes the output buffer in node 5 for the link to node 6 to fill up. Thus, congestion at one point in the network can quickly propagate throughout a region or throughout all of the network. While flow control is indeed a powerful tool, we need to use it in such a way as to manage the traffic on the entire network.

Figure 9.15 shows the effect of congestion in general terms. Figure 9.15a plots the throughput of a network (number of packets delivered to destination stations) versus the offered load (number of packets transmitted by source stations). Both axes are normalized to the maximum capacity of the network, which can be expressed as the rate at which the network is theoretically capable of handling pack-

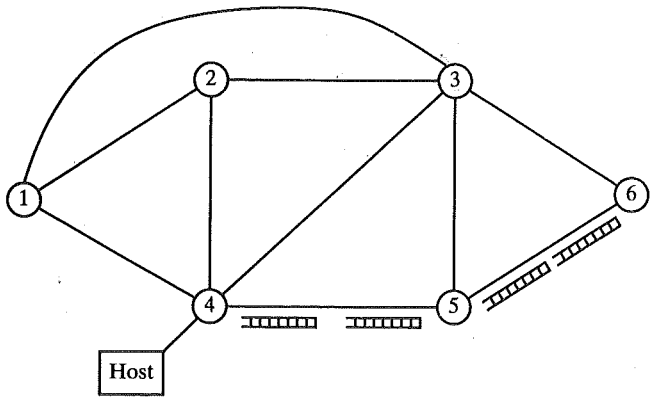


FIGURE 9.14 The interaction of queues in a packet-switching network.

ets. In the ideal case, throughput and, hence, network utilization increase to accommodate an offered load up to the maximum capacity of the network. Utilization then remains at 100%. The ideal case, of course, requires that all stations somehow know the timing and rate of packets that can be presented to the network, which is impossible. If no congestion control is exercised, we have the curve labeled "uncontrolled." As the load increases, utilization increases for a while. Then as the queue lengths at the various nodes begin to grow, throughput actually drops because the buffers at each node are of finite size. When a node's buffers are full, it must discard packets. Thus, the source stations must retransmit the discarded packets in addition to the new packets; this only exacerbates the situation: As more and more packets are retransmitted, the load on the system grows, and more buffers become saturated. While the system is trying desperately to clear the backlog, stations are

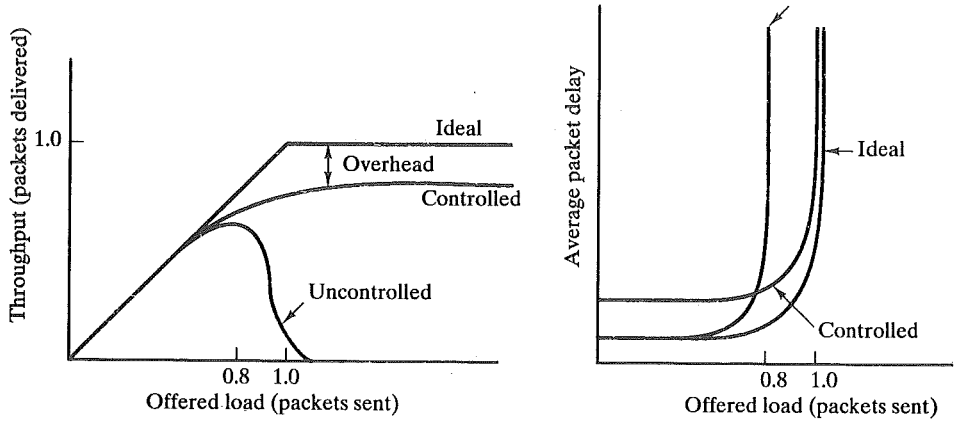


FIGURE 9.15 The effects of congestion.

pumping old and new packets into the system. Even successfully delivered packets may be retransmitted because it takes so long to acknowledge them: The sender assumes that the packet did not go through. Under these circumstances, the effective capacity of the system is virtually zero.

It is clear that these catastrophic events must be avoided; this is the task of congestion control. The object of all congestion-control techniques is to limit queue lengths at the nodes so as to avoid throughput collapse. This control involves some unavoidable overhead. Thus, a congestion-control technique cannot perform as well as the theoretical ideal. However, a good congestion-control strategy will avoid throughput collapse and maintain a throughput that differs from the ideal by an amount roughly equal to the overhead of the control.

Figure 9.15b points out that no matter what technique is used, the average delay experienced by packets grows without bound as the load approaches the capacity of the system. Note that initially the uncontrolled policy results in less delay than a controlled policy, because of its lack of overhead. However, the uncontrolled policy will saturate at lower load.

A number of control mechanisms for congestion control in packet-switching networks have been suggested and tried. The following are examples:

1. Send a control packet from a congested node to some or all source nodes. This choke packet will have the effect of stopping or slowing the rate of transmission from sources and, hence, limit the total number of packets in the network. This approach requires additional traffic on the network during a period of congestion.
2. Rely on routing information. Routing algorithms, such as ARPANETs, provide link delay information to other nodes, which influences routing decisions. This information could also be used to influence the rate at which new packets are produced. Because these delays are being influenced by the routing decision, they may vary too rapidly to be used effectively for congestion control.
3. Make use of an end-to-end probe packet. Such a packet could be time-stamped to measure the delay between two particular endpoints. This procedure has the disadvantage of adding overhead to the network.
4. Allow packet-switching nodes to add congestion information to packets as they go by. There are two possible approaches here. A node could add such information to packets going in the direction opposite of the congestion. This information quickly reaches the source node, which can reduce the flow of packets into the network. Alternatively, a node could add such information to packets going in the same direction as the congestion. The destination either asks the source to adjust the load or returns the signal back to the source in the packets (or acknowledgments) going in the reverse direction.

9.4 X.25

Perhaps the best-known and most widely used protocol standard is X.25, which was originally approved in 1976 and subsequently revised in 1980, 1984, 1988, 1992, and

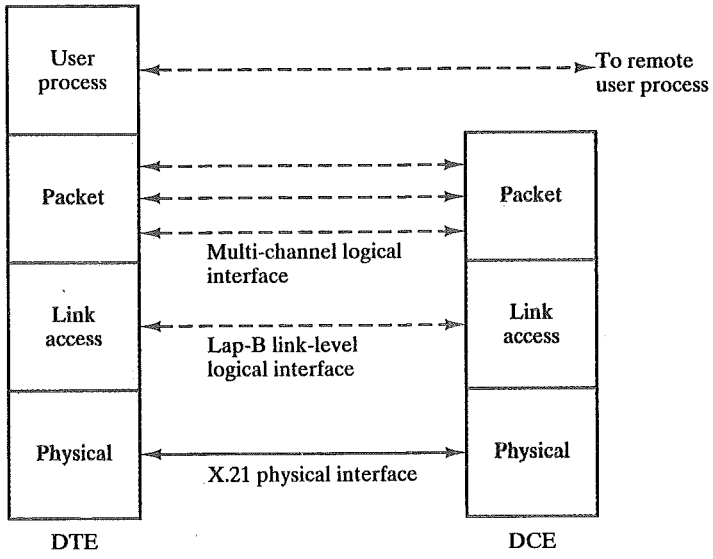


FIGURE 9.16 X.25 interface.

1993. The standard specifies an interface between a host system and a packet-switching network. This standard is almost universally used for interfacing to packet-switching networks and is employed for packet switching in ISDN. In this section, a brief overview of the standard is provided.

The standard specifically calls for three layers of functionality (Figure 9.16):

- Physical layer
- Link layer
- Packet layer

These three layers correspond to the lowest three layers of the OSI model (see Figure 1.10). The physical layer deals with the physical interface between an attached station (computer, terminal) and the link that attaches that station to the packet-switching node. The standard refers to user machines as data terminal equipment (DTE) and to a packet-switching node to which a DTE is attached as data circuit-terminating equipment (DCE). X.25 makes use of the physical-layer specification in a standard known as X.21, but, in many cases, other standards, such as EIA-232, are substituted. The link layer provides for the reliable transfer of data across the physical link by transmitting the data as a sequence of frames. The link-layer standard is referred to as LAPB (Link Access Protocol—Balanced). LAPB is a subset of HDLC, described in Chapter 6. The packet layer provides an external virtual-circuit service, and is described in this section.

Figure 9.17 illustrates the relationship between the levels of X.25. User data are passed down to X.25 level 3, which appends control information as a header, creating a *packet*. This control information is used in the operation of the protocol, as we shall see. The entire X.25 packet is then passed down to the LAPB entity, which appends control information at the front and back of the packet, forming an

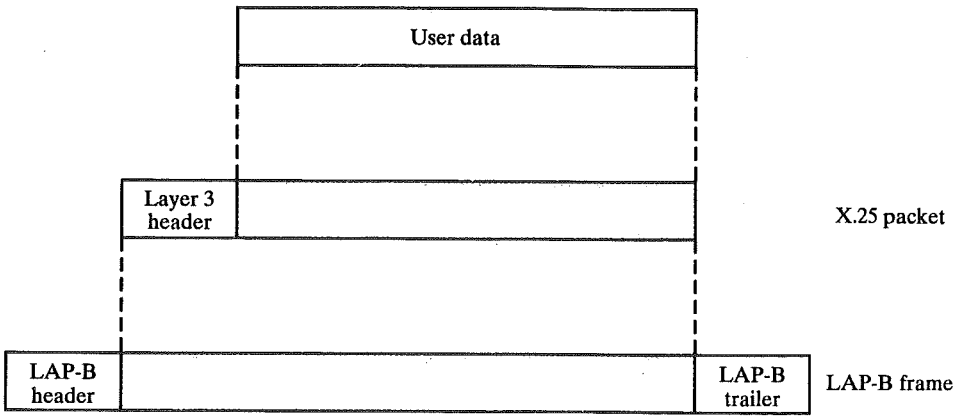


FIGURE 9.17 User data and X.25 protocol control information.

LAPB *frame*. Again, the control information in the frame is needed for the operation of the LAPB protocol.

Virtual Circuit Service

With the X.25 packet layer, data are transmitted in packets over external virtual circuits. The virtual-circuit service of X.25 provides for two types of virtual circuit: virtual call and permanent virtual circuit. A *virtual call* is a dynamically established virtual circuit using a call setup and call clearing procedure, explained below. A *permanent virtual circuit* is a fixed, network-assigned virtual circuit. Data transfer occurs as with virtual calls, but no call setup or clearing is required.

Figure 9.18 shows a typical sequence of events in a virtual call. The left-hand part of the figure shows the packets exchanged between user machine *A* and the packet-switching node to which it attaches; the right-hand part shows the packets exchanged between user machine *B* and its node. The routing of packets inside the network is not visible to the user.

The sequence of events is as follows:

1. A requests a virtual circuit to *B* by sending a Call-Request packet to *A*'s DCE. The packet includes the source and destination addresses, as well as the virtual-circuit number to be used for this new virtual circuit. Future incoming and outgoing transfers will be identified by this virtual-circuit number.
2. The network routes this call request to *B*'s DCE.
3. *B*'s DCE receives the Call Request and sends an Incoming-Call packet to *B*. This packet has the same format as the Call-Request packet but utilizes a different virtual-circuit number, selected by *B*'s DCE from the set of locally unused numbers.

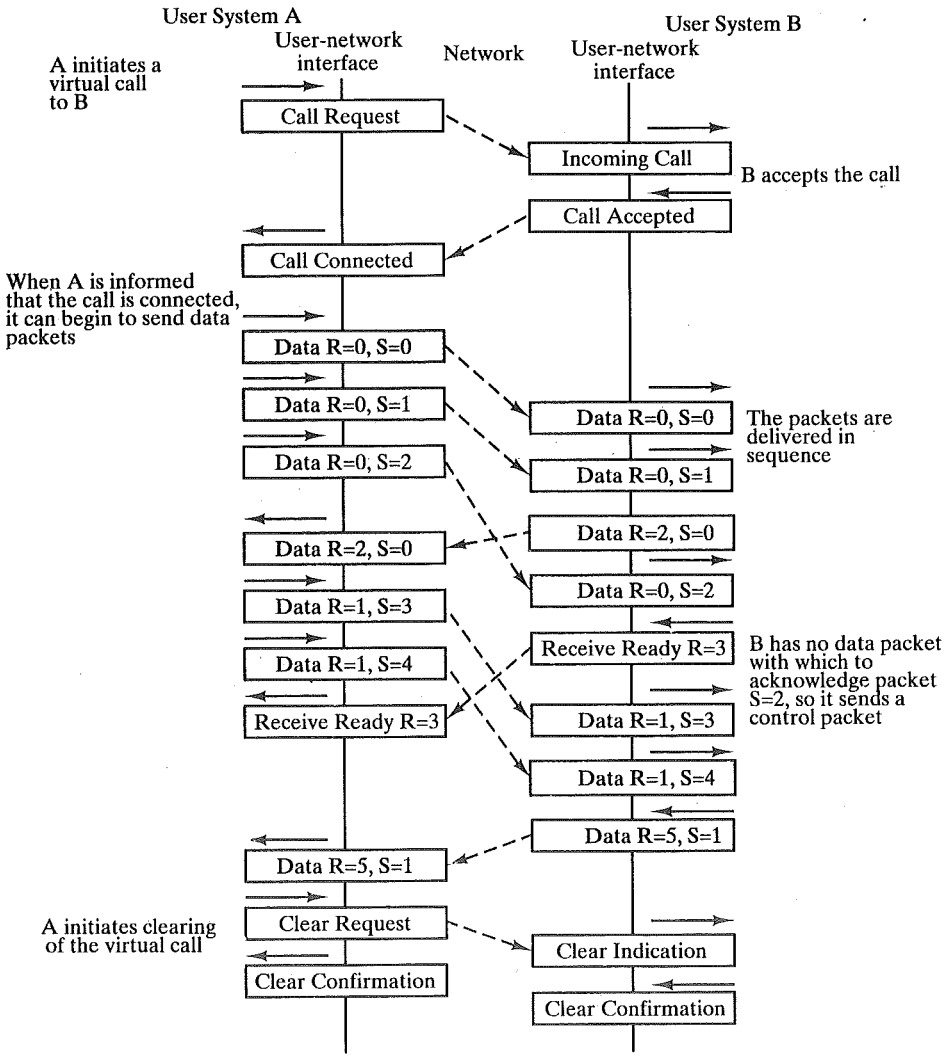


FIGURE 9.18 Sequence of events: X.25 protocol.

4. *B* indicates acceptance of the call by sending a Call-Accepted packet specifying the same virtual circuit number as that of the Incoming-Call packet.
5. *A*'s DCE receives the Call Accepted and sends a Call-Connected packet to *A*. This packet has the same format as the Call-Accepted packet but the same virtual-circuit number as that of the original Call-Request packet.
6. *A* and *B* send data and control packets to each other using their respective virtual-circuit numbers.

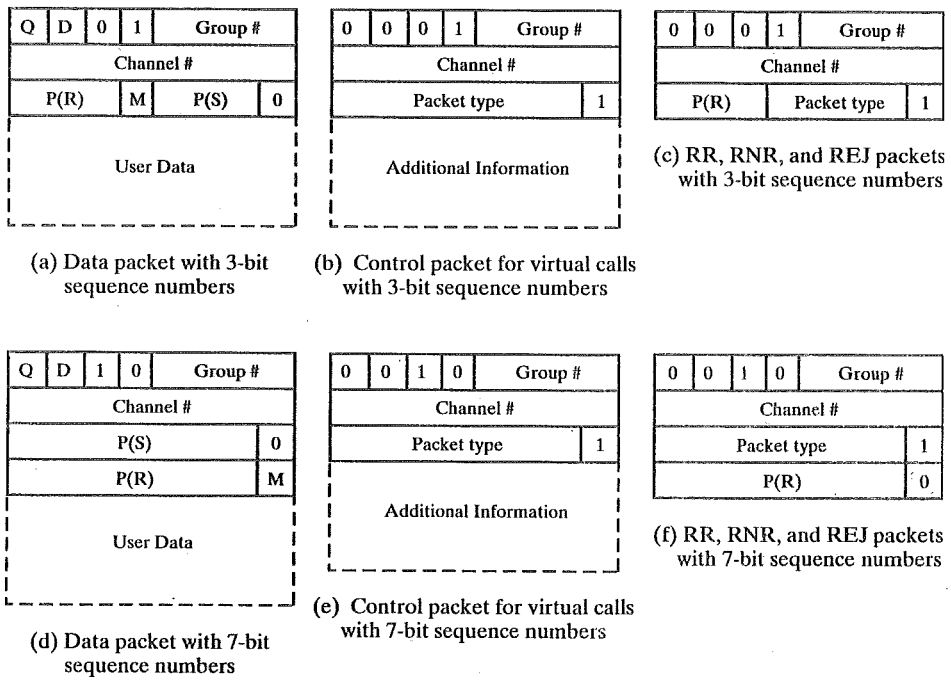
7. *A* (or *B*) sends a Clear-Request packet to terminate the virtual circuit and receives a Clear-Confirmation packet.
8. *B* (or *A*) receives a Clear-Indication packet and transmits a Clear-Confirmation packet.

We now turn to some of the details of the standard.

Packet Format

Figure 9.19 shows the basic X.25 packet formats. For user data, the data are broken up into blocks of some maximum size, and a 24-bit or 32-bit header is appended to each block to form a **data packet**. The header includes a 12-bit virtual-circuit number (expressed as a 4-bit group number and an 8-bit channel number). The P(S) and P(R) fields support the functions of flow control and error control on a virtual-circuit basis, as explained below. The M and D bits are described below. The Q bit is not defined in the standard, but allows the user to distinguish two types of data.

In addition to transmitting user data, X.25 must transmit control information related to the establishment, maintenance, and termination of virtual circuits. Control information is transmitted in a **control packet**. Each control packet includes the virtual-circuit number, the packet type, which identifies the particular control function, as well as additional control information related to that function. For example, a Call-Request packet includes the following additional fields:



(a) Data packet with 3-bit sequence numbers

(b) Control packet for virtual calls with 3-bit sequence numbers

(c) RR, RNR, and REJ packets with 3-bit sequence numbers

(d) Data packet with 7-bit sequence numbers

(e) Control packet for virtual calls with 7-bit sequence numbers

(f) RR, RNR, and REJ packets with 7-bit sequence numbers

FIGURE 9.19 X.25 packet formats.

- Calling DTE address length (4 bits): length of the corresponding address field in 4-bit units.
- Called DTE address length (4 bits): length of the corresponding address field in 4-bit units.
- DTE addresses (variable): the calling and called DTE addresses.
- Facilities: a sequence of facility specifications. Each specification consists of an 8-bit facility code and zero or more parameter codes. An example of a facility is reverse charging.

Table 9.3 lists all of the X.25 packets. Most of these have already been discussed. A brief description of the remainder follow.

A DTE may send an Interrupt packet that bypasses the flow-control procedures for data packets. The interrupt packet is to be delivered to the destination DTE by the network at a higher priority than data packets in transit. An example of the use of this capability is the transmission of a terminal-break character.

The Reset packets provide a facility for recovering from an error by reinitializing a virtual circuit, meaning that the sequence numbers on both ends are set to 0. Any data or interrupt packets in transit are lost. A reset can be triggered by a number of error conditions, including loss of a packet, sequence number error, congestion, or loss of the network's internal logical connection. In the latter case, the two DCEs must rebuild the internal logical connection to support the still-existing X.25 DTE-DTE virtual circuit.

A more serious error condition is dealt with by a Restart, which terminates all active virtual calls. An example of a condition warranting restart is temporary loss of access to the network.

The Diagnostic packet provides a means to signal certain error conditions that do not warrant reinitialization. The Registration packets are used to invoke and confirm X.25 facilities.

Multiplexing

Perhaps the most important service provided by X.25 is multiplexing. A DTE is allowed to establish up to 4095 simultaneous virtual circuits with other DTEs over a single physical DTE-DCE link. The DTE can internally assign these circuits in any way it pleases. Individual virtual circuits could correspond to applications, processes, or terminals, for example. The DTE-DCE link provides full-duplex multiplexing; that is, at any time, a packet associated with a given virtual circuit can be transmitted in either direction.

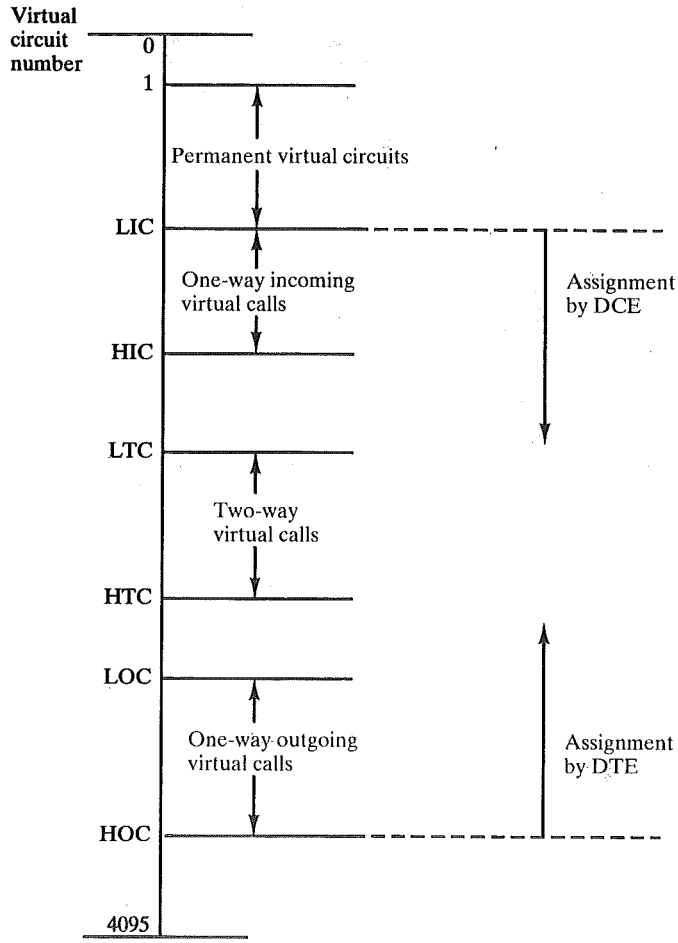
To sort out which packets belong to which virtual circuits, each packet contains a 12-bit virtual-circuit number (expressed as a 4-bit logical group number plus an 8-bit logical channel number). The assignment of virtual-circuit numbers follows the convention depicted in Figure 9.20. Number zero is always reserved for diagnostic packets common to all virtual circuits. Then, contiguous ranges of numbers are allocated for four categories of virtual circuits. Permanent virtual circuits are assigned numbers beginning with 1. The next category is one-way, incoming virtual calls. This means that only incoming calls from the network can be assigned these

TABLE 9.3 X.25 Packet types and parameters.

Packet type		Service		Parameters
From DTE to DCE	From DCE to DTE	VC	PVC	
Call setup and clearing				
Call request	Incoming call	X		Calling DTE address, called DTE address, facilities, call user data
Call accepted	Call connected	X		Calling DTE address, called DTE address, facilities, call user data
Clear request	Clear indication	X		Clearing cause, diagnostic code, calling DTE address, called DTE address, facilities, clear user data
Clear confirmation	Clear confirmation	X		Calling DTE address, called DTE address, facilities
Data and interrupt				
Data	Data	X	X	—
Interrupt	Interrupt	X	X	Interrupt user data
Interrupt confirmation	Interrupt confirmation	X	X	—
Flow Control and Reset				
RR	RR	X	X	P(R)
RNR	RNR	X	X	P(R)
REJ		X	X	P(R)
Reset request	Reset indication	X	X	Resetting cause, diagnostic code
Reset confirmation	Reset confirmation	X	X	—
Restart				
Restart request	Restart indication	X	X	Restarting cause, diagnostic code
Restart confirmation	Restart confirmation	X	X	—
Diagnostic				
	Diagnostic	X	X	Diagnostic code, diagnostic explanation
Registration				
Registration request		X	X	DTE address, DCE address, registration
	Registration Confirmation	X	X	Cause, diagnostic, DTE address, DCE address, registration

numbers; the virtual circuit, however, is two-way (full duplex). When a call request comes in, the DCE selects an unused number from this category.

One-way outgoing calls are those initiated by the DTE. In this case, the DTE selects an unused number from among those allocated for these calls. This separa-



LEGEND

LIC = Lowest incoming channel	HTC = Highest two-way channel	Virtual circuit number =
HIC = Highest incoming channel	LOC = Lowest outgoing channel	logical group number and
LTC = Lowest two-way channel	HOC = Highest outgoing channel	logical channel number

FIGURE 9.20 Virtual-circuit number assignment.

tion of categories is intended to avoid the simultaneous selection of the same number for two different virtual circuits by the DTE and DCE.

The two-way virtual-call category provides an overflow for allocation shared by DTE and DCE, allowing for peak differences in traffic flow.

Flow and Error Control

Flow control and error control at the X.25 packet layer are virtually identical in format and procedure to flow control used for HDLC, as described in Chapter 6.

A sliding-window protocol is used. Each data packet includes a send sequence number, P(S), and a receive sequence number, P(R). As a default, 3-bit sequence numbers are used. Optionally, a DTE may request, via the user-facility mechanism, the use of extended 7-bit sequence numbers. As Figure 9.19 indicates, for 3-bit sequence numbers, the third and fourth bits of all data and control packets are 01; for 7-bit sequence numbers, the bits are 10.

P(S) is assigned by the DTE on outgoing packets on a virtual circuit basis; that is, the P(S) of each new outgoing data packet on a virtual circuit is one more than that of the preceding packet, modulo 8 or modulo 128. P(R) contains the number of the next packet expected from the other side of a virtual circuit; this provides for piggybacked acknowledgment. If one side has no data to send, it may acknowledge incoming packets with the Receive-Ready (RR) and Receive-not-Ready (RNR) control packets, with the same meaning as for HDLC. The default window size is 2, but it may be set as high as 7 for 3-bit sequence numbers and as high as 127 for 7-bit sequence numbers.

Acknowledgment (in the form of the P(R) field in the data, RR, or RNR packet), and hence flow control, may have either local or end-to-end significance, based on the setting of the D bit. When D = 0, (the usual case), acknowledgment is exercised between the DTE and the network. This communication is used by the local DCE and/or the network to acknowledge receipt of packets and to control the flow from the DTE into the network. When D = 1, acknowledgments come from the remote DTE.

The basic form of *error control* is go-back-N ARQ. Negative acknowledgment is in the form of a Reject (REJ) control packet. If a node receives a negative acknowledgment, it will retransmit the specified packet and all subsequent packets.

Packet Sequences

X.25 provides the capability, called a *complete packet sequence*, to identify a contiguous sequence of data packets. This feature has several uses. One important use is by internetworking protocols (described in Part III) to allow longer blocks of data to be sent across a network with a smaller packet-size restriction without losing the integrity of the block.

To specify this mechanism, X.25 defines two types of packets: A packets and B packets. An *A packet* is one in which the M bit is set to 1, the D bit is set to 0, and the packet is full (equal to the maximum allowable packet length). A *B packet* is any packet that is not an A packet. A complete packet sequence consists of zero or more A packets followed by a B packet. The network may combine this sequence to make a larger packet. The network may also segment a B packet into smaller packets to produce a complete packet sequence.

The way in which the B packet is handled depends on the setting of the M and D bits. If D = 1, an end-to-end acknowledgment is sent by the receiving DTE to the sending DTE. This is, in effect, an acknowledgment of the entire complete packet sequence. If M = 1, there are additional complete packet sequences to follow. This enables the formation of subsequences as part of a larger sequence, so that end-to-end acknowledgment can occur before the end of the larger sequence.

Figure 9.21 shows examples of these concepts. It is the responsibility of the

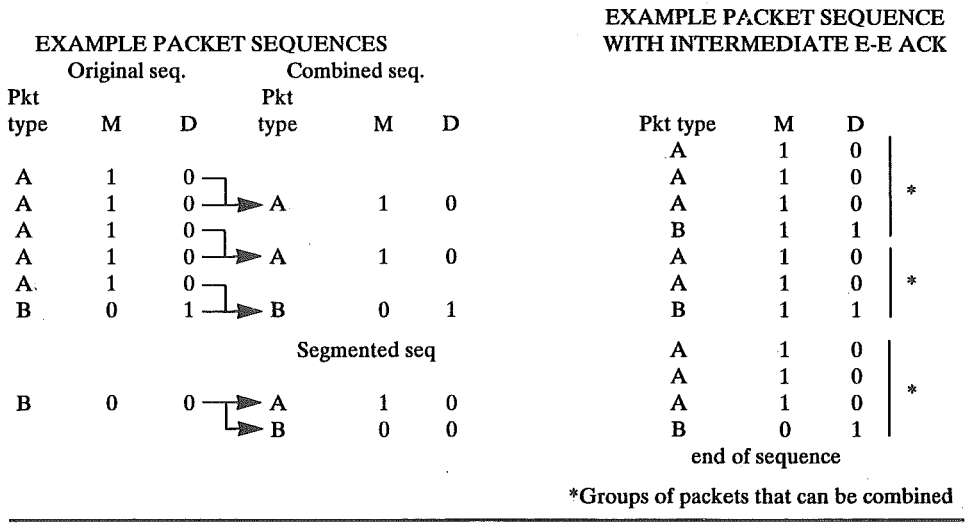


FIGURE 9.21 X.25 packet sequences.

DCEs to reconcile the changes in sequence numbering that segmentation and reassembly cause.

9.5 RECOMMENDED READING

The literature on packet switching is enormous. Only a few of the worthwhile references are mentioned here. Books with good treatments of this subject include [SPOH93], [BERT92] and [SPRA91]. There is also a large body of literature on performance; good summaries are to be found in [STUC85], [SCHW77], and [KLEI76].

BERT92 Bertsekas, D. and Gallager, R. *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.

KLEI76 Kleinrock, L. *Queuing Systems, Volume II: Computer Applications*. New York: Wiley, 1976.

SCHW77 Schwartz, M. *Computer-Communication Network Design and Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1977.

SPOH93 Spohn, D. *Data Network Design*. New York: McGraw-Hill, 1994.

SPRA91 Spragins, J., Hammond, J., and Pawlikowski, K. *Telecommunications Protocols and Design*. Reading, MA.: Addison-Wesley, 1991.

STUC85 Stuck, B. and Arthurs, E. *A Computer Communications Network Performance Analysis Primer*. Englewood Cliffs, NJ: Prentice Hall, 1985.

9.6 PROBLEMS

9.1 Explain the flaw in the following logic:
Packet switching requires control and address bits to be added to each packet. This causes considerable overhead in packet switching. In circuit switching, a transparent circuit is established. No extra bits are needed.
a. Therefore, there is no overhead in circuit switching.

b. Because there is no overhead in circuit switching, line utilization must be more efficient than in packet switching.

9.2 Define the following parameters for a switching network:

N = number of hops between two given end systems
L = message length in bits
B = data rate, in bits per second (bps), on all links
P = packet size
H = overhead (header) bits per packet
S = call setup time (circuit switching or virtual circuit) in seconds
D = propagation delay per hop in seconds

a. For $N = 4$, $l = 3200$, $b = 9600$, $P = 1024$, $H = 16$, $S = 0.2$, $D = 0.001$, compute the end-to-end delay for circuit switching, virtual-circuit packet switching, and datagram packet switching. Assume that there are no acknowledgments.
b. Derive general expressions for the three techniques of part (a), taken two at a time (six expressions in all), showing the conditions under which the delays are equal.

9.3 What value of P, as a function of N, B, and H, results in minimum end-to-end delay on a datagram network? Assume that L is much larger than P, and D is zero.

9.4 Consider a packet-switching network of N nodes, connected by the following topologies:
a. Star: one central node with no attached station; all other nodes attach to the central node.
b. Loop: each node connects to two other nodes to form a closed loop.
c. Fully connected: each node is directly connected to all other nodes.

For each case, give the average number of hops between stations.

9.5 Consider a binary tree topology for a packet-switching network. The root node connects to two other nodes. All intermediate nodes connect to one node in the direction toward the root, and two in the direction away from the root. At the bottom are nodes with just one link back toward the root. If there are $2^N - 1$ nodes, derive an expression for the mean number of hops per packet for large N, assuming that trips between all node pairs are equally likely.

9.6 Dijkstra's algorithm, for finding the least-cost path from a specified node s to a specified node t, can be expressed in the following program:

```
for n := 1 to N do
    begin
        D[n] := ∞; final[n] := false; {all nodes are temporarily labeled with ∞}
        pred[n] := 1
    end;
D[s] := 0; final[s] := true;          {node s is permanently labeled with 0}
recent := s;                          {the most recent node to be permanently labeled is s}
path := true;
{initialization over }

while final[t] = false do
begin
    for n := 1 to N do  {find new label}
        if (d[recent, n] <∞) AND (NOT final[n]) then
        {for every immediate successor of recent that is not permanently labeled, do }
            begin {update temporary labels}
                newlabel := D[recent] + d[recent,n];
                if newlabel <D[n] then
                        begin D[n] := newlabel; pred[n] := recent end
                        {re-label n if there is a shorter path via node recent and make
                          recent the predecessor of n on the shortest path from s}
            end;
    temp := ∞;
```

```
for x := 1 to N do {find node with smallest temporary label}
    if (NOT final[x]) AND (D[x] <temp) then
            begin y := x; temp: =D[x] end;
if temp < ∞ then {there is a path} then
    begin final[y] := true; recent := y end
    {y, the next closest node to s gets permanently labeled}
else begin path := false; final[t] := true end
end
```

In this program, each node is assigned a temporary label initially. As a final path to a node is determined, it is assigned a permanent label equal to the cost of the path from *s*. Write a similar program for the Bellman-Ford algorithm. Hint: The Bellman-Ford algorithm is often called a label-correcting method, in contrast to Dijkstra's label-setting method.

9.7 In the discussion of Dijkstra's algorithm in Appendix 9A, it is asserted that at each iteration, a new node is added to M and that the least-cost path for that new node passes only through nodes already in M. Demonstrate that this is true. Hint: Begin at the beginning. Show that the first node added to M must have a direct link to the source node. Then show that the second node to M must either have a direct link to the source node or a direct link to the first node added to M, and so on. Remember that all link costs are assumed nonnegative.

9.8 In the discussion of the Bellman-Ford algorithm in Appendix 9A, it is asserted that at the iteration for which $H = K$, if any path of length $K + 1$ is defined, the first K hops of that path form a path defined in the previous iteration. Demonstrate that this is true.

9.9 In step 3 of Dijkstra's algorithm, the least-cost path values are only updated for nodes not yet in M. Is it not possible that a lower-cost path could be found to a node already in M? If so, demonstrate by example. If not, provide reasoning as to why not.

9.10 Using Dijkstra's algorithm, generate a least-cost route to all other nodes for nodes 2 through 6 of Figure 9.5. Display the results as in Table 9.4a; do the same for the Bellman-Ford algorithm.

9.11 Apply Dijkstra's routing algorithm to the networks in Figure 9.22 (next page). Provide a table similar to Table 9.4 and a figure similar to Figure 9.9.

9.12 Repeat Problem 9.11 using the Bellman-Ford algorithm.

9.13 Will Dijkstra's algorithm and the Bellman-Ford algorithm always yield the same solutions? Why or why not?

9.14 Both Dijkstra's algorithm and the Bellman-Ford algorithm find the least-cost paths from one node to all other nodes. The Floyd-Warshall algorithm finds the least-cost paths between all pairs of nodes together. Define

N = set of nodes in the network
d_{ij} = link cost from node i to node j; $d_{ii} = 0$; and $d_{ij} = \infty$ if the nodes are not directly connected
$D^{(n)}_{ij}$ = cost of the least-cost path from node i to node j with the constraint that only nodes $1, 2, \ldots, n$ can be used as intermediate nodes on paths

The algorithm has the following steps:

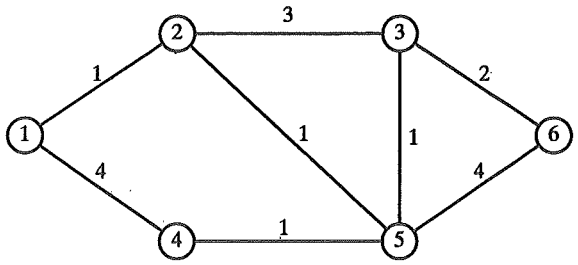
1. Initialize:

$$D^{(0)}_{ij} = d_{ij}, \text{ for all } i, j, i \neq j$$

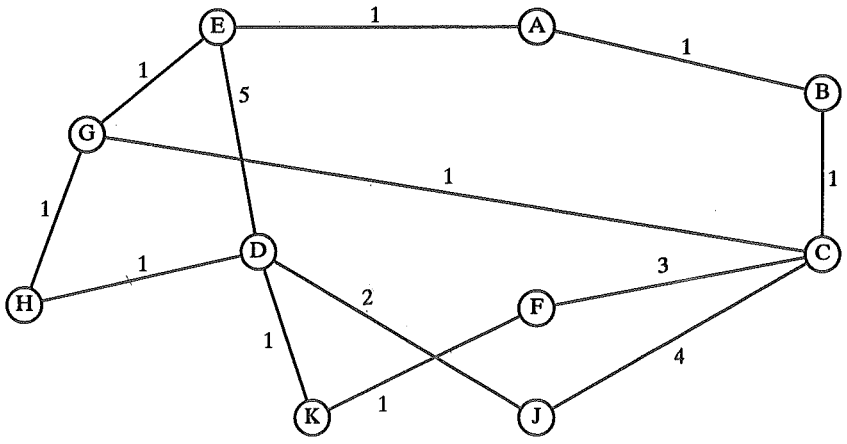
2. For $n = 0, 1, \ldots, N-1$

$$D^{(n+1)}_{ij} = \text{Min}[D^{(n)}_{ij}, D^{(n)}_{i(n+1)} + D^{(n)}_{(n+1)j}] \text{ for all } i \neq j$$

Explain the algorithm in words. Use induction to demonstrate that the algorithm works.



(a)



(b)

FIGURE 9.22 Example of least-cost routing algorithm (based on Table 9.4).

9.15 In Figure 9.8, node 1 sends a packet to node 6 using flooding. Counting the transmission of one packet across one link as a load of one, what is the total load generated if
a. Each node discards duplicate incoming packets?
b. A hop count field is used and is initially set to 5?

9.16 It was shown that flooding can be used to determine the minimum-hop route. Can it be used to determine the minimum delay route?

9.17 With random routing, only one copy of the packet is in existence at a time. Nevertheless, it would be wise to utilize a hop count field. Why?

9.18 Another adaptive routing scheme is known as backward learning. As a packet is routed through the network, it carries not only the destination address, but the source address plus a running hop count that is incremented for each hop. Each node builds a routing table that gives the next node and hop count for each destination. How is the packet information used to build the table? What are the advantages and disadvantages of this technique?

9.19 Build a centralized routing directory for the networks of Problem 9.11.

9.20 Consider a system using flooding with a hop counter. Suppose that the hop counter is originally set to the "diameter" of the network. When the hop count reaches zero, the packet is discarded, except at its destination. Does this procedure always ensure that a

packet will reach its destination if there exists at least one operable path? Why or why not?

9.21 Assuming no malfunction in any of the stations or nodes of a network, is it possible for a packet to be delivered to the wrong destination?

9.22 Flow-control mechanisms are used at both levels 2 and 3 of X.25. Are both necessary, or is this redundant? Explain.

9.23 There is no error-detection mechanism (frame check sequence) in X.25. Isn't this needed to assure that all of the packets are delivered properly?

9.24 When an X.25 DTE and the DCE to which it attaches both decide to put a call through at the same time, a call collision occurs and the incoming call is canceled. When both sides try to clear the same virtual circuit simultaneously, the clear collision is resolved without canceling either request; the virtual circuit in question is cleared. Do you think that simultaneous resets are handled like call collisions or clear collisions? Why?

9.25 In X.25, why is the virtual-circuit number used by one station of two communicating stations different from the virtual-circuit number used by the other station? After all, it is the same full-duplex virtual circuit.

9A APPENDIX

LEAST-COST ALGORITHMS

VIRTUALLY ALL PACKET-SWITCHED networks base their routing decision on some form of least-cost criterion. If the criterion is to minimize the number of hops, each link has a value of 1. More typically, the link value is inversely proportional to the link capacity, proportional to the current load on the link, or some combination of the two. In any case, these link or hop costs are used as input to a least-cost routing algorithm, which can be simply stated as follows:

> Given a network of nodes connected by bidirectional links, where each link has a cost associated with it in each direction, define the cost of a path between two nodes as the sum of the costs of the links traversed. For each pair of nodes, find the path with the least cost.

Note that the cost of a link may differ in its two directions; this would be true, for example, if the cost of a link equaled the length of the queue of packets awaiting transmission from each of the two nodes on the link.

Most least-cost routing algorithms in use in packet-switched networks are variations of one of two common algorithms, known as Dijkstra's algorithm and the Bellman-Ford algorithm.[1] This appendix provides a summary of these two algorithms.

Dijkstra's Algorithm

Dijkstra's algorithm [DIJK59] can be stated as: Find the shortest paths from a given source node to all other nodes by developing the paths in order of increasing path length. The algorithm proceeds in stages. By the kth stage, the shortest paths to the k nodes closest to (least cost away from) the source node have been determined; these nodes are in a set M. At stage $(k + 1)$, the node not in M that has the shortest path from the source node is added to M. As each node is added to M, its path from the source is defined. The algorithm can be formally described as follows. Use the following definitions:

N = set of nodes in the network
s = source node
M = set of nodes so far incorporated by the algorithm
d_{ij} = link cost from node i to node j; $d_{ii} = 0$; $d_{ij} = \infty$ if the two nodes are not directly connected; $d_{ij} \geq 0$ if the two nodes are directly connected
D_n = cost of the least-cost path from node s to node n that is currently known to the algorithm

The algorithm has three steps; steps 2 and 3 are repeated until $M = N$. That is, steps 2 and 3 are repeated until final paths have been assigned to all nodes in the network:

1. Initialize:
   $M = \{s\}$ (i.e., the set of nodes so far incorporated consists of only the source node)
   $D_n = d_{sn}$ for $n \neq s$ (i.e., the initial-path costs to neighboring nodes are simply the link costs)
2. Find the neighboring node not in M that has the least-cost path from node s and incorporate that node into M: This can be expressed as

[1] As we shall see in Part III, this statement is also true of routing in internetworks.

Find $w \notin M$ such that $D_w = \min_{j \notin M} D_j$
Add w to M

3. Update least-cost paths:
$D_n = \min[D_n, D_w + d_{wn}]$ for all $n \notin M$
If the latter term is the minimum, the path from s to n is now the path from s to w, concatenated with the link from w to n.

One iteration of steps 2 and 3 adds one new node to M and defines the least-cost path from s to that node. That path passes only through nodes that are in M; to see this, consider the following line of reasoning. After k iterations, there are k nodes in M, and the least-cost path from s to each of these nodes has been defined. Now consider all possible paths from s to nodes not in M. Among those paths, there is one of least cost that passes exclusively through nodes in M (see Problem 9.7), ending with a direct link from some node in M to a node not in M. This node is added to M, and the associated path is defined as the least-cost path for that node.

Table 9.4a shows the result of applying this algorithm to Figure 9.5, using $s = 1$. Note that at each step, the path to each node plus the total cost of that path is generated. After the final iteration, the least-cost path to each node and the cost of that path have been developed. The same procedure can be used with node 2 as source node, and so on.

Bellman-Ford Algorithm

The Bellman-Ford algorithm [FORD62] can be stated as follows: Find the shortest paths from a given source node subject to the constraint that the paths contain, at most, one link; then find the shortest paths with a constraint of paths of, at most, two links, and so on. This algorithm also proceeds in stages. The algorithm can be formally described as follows. Use the following definitions:

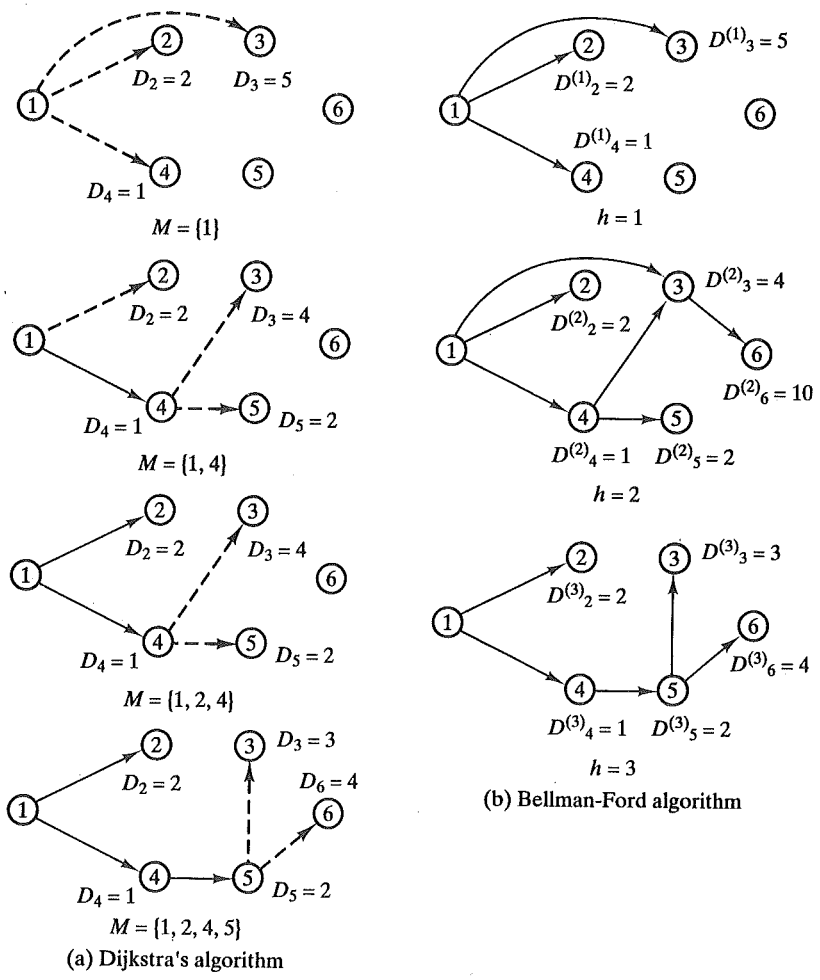
s = source node
d_{ij} = link cost from node i to node j; $d_{ii} = 0$; $d_{ij} = \infty$ if the two nodes are not directly connected; $d_{ij} \geq 0$ if the two nodes are directly connected
h = maximum number of links in a path at the current stage of the algorithm
$D_n^{(h)}$ = cost of the least-cost path from node s to node n under the constraint of no more than h links

The algorithm has the following steps, step 2 of which is repeated until none of the costs change:

1. Initialize:
$$D_n^{(0)} = \infty, \text{ for all } n \neq s$$
$$D_s^{(h)} = 0, \text{ for all } h$$

2. For each successive $h \geq 0$:
$$D_n^{(h+1)} = \min_j [D_j^{(h)} + d_{jn}]$$
The path from s to i terminates with the link from j to i.

For the iteration of step 2 with $h = K$, and for each destination node n, the algorithm compares potential paths from s to n of length $K + 1$ with the path that existed at the end of the previous iteration. If the previous, shorter, path has less cost, then that path is retained. Otherwise, a new path with length $K + 1$ is defined from s to n; this path consists of a path of length K from s to some node j, plus a direct hop from node j to node n. In this case, the path from s to j that is used is the K-hop path for j defined in the previous iteration (see Problem 9.8).



(a) Dijkstra's algorithm

(b) Bellman-Ford algorithm

FIGURE 9.23 Caption to come.

Table 9.4b shows the result of applying this algorithm to Figure 9.5, using $s = 1$. At each step, the least-cost paths with a maximum number of links equal to h are found. After the final iteration, the least-cost path to each node, and the cost of that path, have been developed. The same procedure can be used with node 2 as source node, and so on. Note that the results agree with those obtained using Dijkstra's algorithm. Figure 9.23 illustrate the results of Table 9.4.

Comparison

One interesting comparison can be made between these two algorithms, having to do with what information needs to be gathered. Consider first the Bellman-Ford algorithm. In step 2, the calculation for node n involves knowledge of the link cost to all neighboring nodes to node n (d_{jn}) plus the total path cost to each of those neighboring nodes from a particular source node s ($D_j^{(h)}$). Each node can maintain a set of costs and associated paths for every other node in the network, and can exchange this information with its direct neighbors from time to time. Each node can therefore use the expression in step 2 of the Bellman-Ford algorithm, based only on information from its neighbors and knowledge of its link costs, to

TABLE 9.4 Example of least-cost routing algorithms (using Figure 9.5).

(a) Dijkstra's Algorithm (s = 1)

Iteration	M	D_2	Path	D_3	Path	D_4	Path	D_5	Path	D_6	Path
1	{1}	2	1-2	5	1-3	1	1-4	∞	—	∞	—
2	{1, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	—
3	{1, 2, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	—
4	{1, 2, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1, 2, 3, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

(b) Bellman-Ford Algorithm (s = 1)

h	$D_2^{(h)}$	Path	$D_3^{(h)}$	Path	$D_4^{(h)}$	Path	$D_5^{(h)}$	Path	$D_6^{(h)}$	Path
0	∞	—	∞	—	∞	—	∞	—	∞	—
1	2	1-2	5	1-3	1	1-4	∞	—	∞	—
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

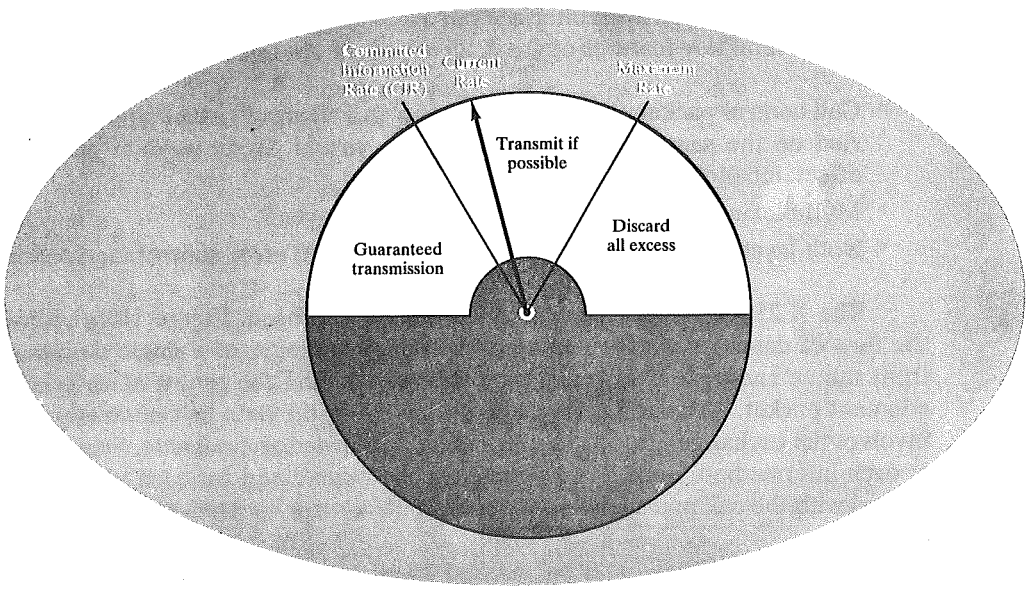
update its costs and paths. On the other hand, consider Dijkstra's algorithm. Step 3 appears to require that each node must have complete topological information about the network. That is, each node must know the link costs of all links in the network. Thus, for this algorithm, information must be exchanged with all other nodes.

Evaluation of the relative merits of the two algorithms should be done with respect to the desirable attributes listed earlier. The evaluation will depend on the implementation approach and on the specific implementation.

A final point: Both algorithms are known to converge under static conditions of topology and link costs and will converge to the same solution. If the link costs change over time, the algorithm will attempt to catch up with these changes. However, if the link cost depends on traffic, which in turn depends on the routes chosen, then a feedback condition exists, and instabilities may result.

CHAPTER 10

FRAME RELAY



The most important technical innovation to come out of the standardization work on ISDN is frame relay. Although designed for ISDN, frame relay now enjoys widespread use in a variety of public and private networks that do not follow the ISDN standards.

Frame relay represents a significant advance over traditional packet switching and X.25. We begin the chapter with an overview of the differences between these two approaches. Next, the details of the frame relay scheme are examined. Then, the key issue of congestion control in frame relay networks is discussed. For a discussion of ISDN, see Appendix A.

10.1 BACKGROUND

The traditional approach to packet switching makes use of X.25, which not only determines the user-network interface but also influences the internal design of the network. Several key features of the X.25 approach are as follows:

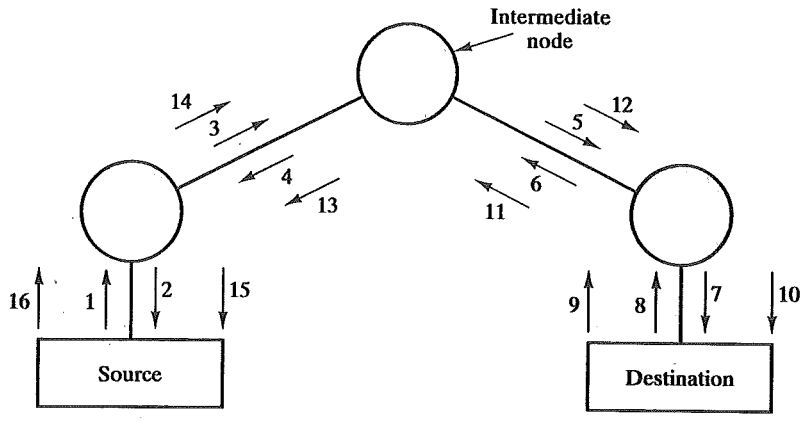
- Call control packets, used for setting up and clearing virtual circuits, are carried on the same channel and the same virtual circuit as data packets. In effect, inband signaling is used.
- Multiplexing of virtual circuits takes place at layer 3.
- Both layer 2 and layer 3 include flow control and error control mechanisms.

The X.25 approach results in considerable overhead. Figure 10.1a indicates the flow of data link frames required for the transmission of a single data packet from source end system to destination end system, and the return of an acknowledgment packet. At each hop through the network, the data link control protocol involves the exchange of a data frame and an acknowledgment frame. Furthermore, at each intermediate node, state tables must be maintained for each virtual circuit to deal with the call management and flow control/error control aspects of the X.25 protocol.

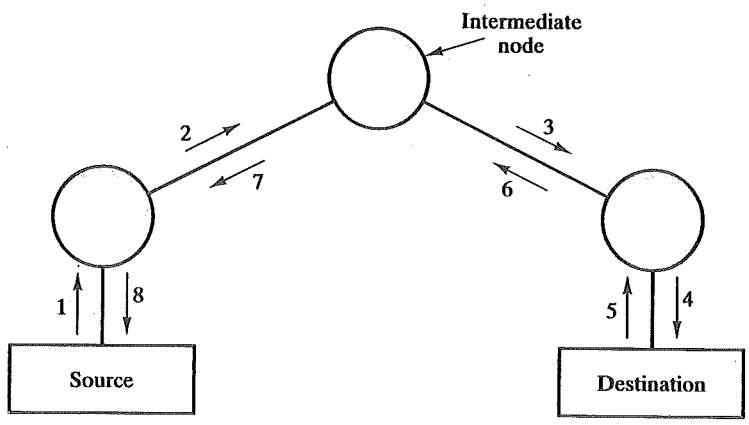
All of this overhead may be justified when there is a significant probability of error on any of the links in the network. This approach may not be the most appropriate for modern digital communication facilities. Today's networks employ reliable digital-transmission technology over high-quality, reliable transmission links, many of which are optical fiber. In addition, with the use of optical fiber and digital transmission, high data rates can be achieved. In this environment, the overhead of X.25 is not only unnecessary, but degrades the effective utilization of the available high data rates.

Frame relaying is designed to eliminate much of the overhead that X.25 imposes on end user systems and on the packet-switching network. The key differences between frame relaying and a conventional X.25 packet-switching service are:

- Call control signaling is carried on a separate logical connection from user data. Thus, intermediate nodes need not maintain state tables or process messages relating to call control on an individual per-connection basis.



(a) Packet-switching network



(b) Frame relay network

FIGURE 10.1 Packet switching versus frame relay: source sending, destination responding.

- Multiplexing and switching of logical connections take place at layer 2 instead of layer 3, eliminating one entire layer of processing.
- There is no hop-by-hop flow control and error control. End-to-end flow control and error control, if they are employed at all, are the responsibility of a higher layer.

Figure 10.1b indicates the operation of frame relay, in which a single-user data frame is sent from source to destination, and an acknowledgment, generated at a higher layer, is carried back in a frame.

Let us consider the advantages and disadvantages of this approach. The principal potential disadvantage of frame relaying, compared to X.25, is that we have lost the ability to do link-by-link flow and error control. (Although frame relay does not provide end-to-end flow and error control, this is easily provided at a higher

layer.) In X.25, multiple virtual circuits are carried on a single physical link, and LAPB is available at the link level for providing reliable transmission from the source to the packet-switching network and from the packet-switching network to the destination. In addition, at each hop through the network, the link control protocol can be used for reliability. With the use of frame relaying, this hop-by-hop link control is lost. However, with the increasing reliability of transmission and switching facilities, this is not a major disadvantage.

The advantage of frame relaying is that we have streamlined the communications process. The protocol functionality required at the user-network interface is reduced, as is the internal network processing. As a result, lower delay and higher throughput can be expected. Studies indicate an improvement in throughput using frame relay, compared to X.25, of an order of magnitude or more [HARB92]. The ITU-T Recommendation I.233 indicates that frame relay is to be used at access speeds up to 2 Mbps.

The ANSI standard T1.606 lists four examples of applications that would benefit from the frame relay service used over a high-speed H channel:

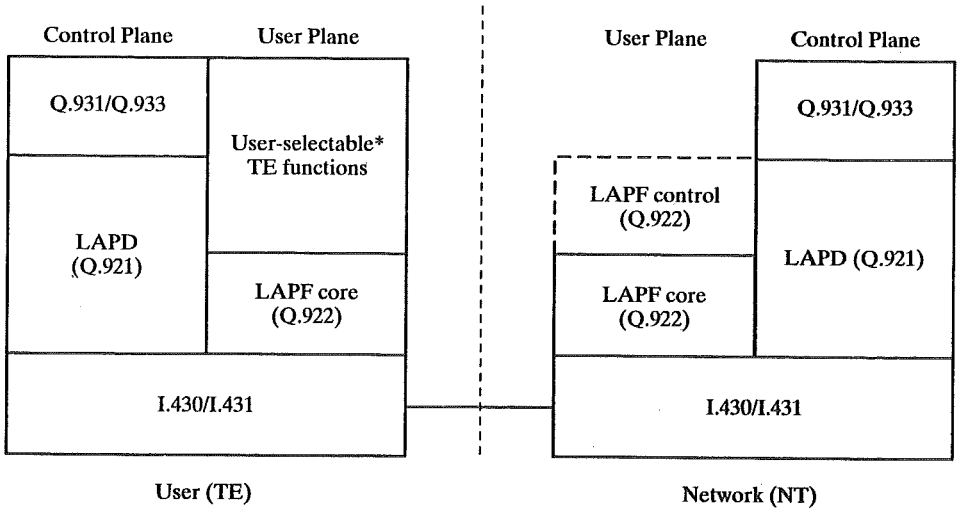
1. *Block-interactive data applications*: An example of a block-interactive application would be high-resolution graphics (e.g., high-resolution videotex, CAD/CAM). The pertinent characteristics of this type of application are low delays and high throughput.
2. *File transfer*: The file transfer application is intended to cater to large file transfer requirements. Transit delay is not as critical for this application as it is, for example, in the first application. High throughput might be necessary in order to produce reasonable transfer times for large files.
3. *Multiplexed low-bit rate*: The multiplexed low-bit-rate application exploits the multiplexing capability of the frame-relaying service in order to provide an economical access arrangement for a large group of low-bit-rate applications. An example of one such low-bit-rate application is given in (4) below. The low-bit-rate sources may be multiplexed onto a channel by an NT function.
4. *Character-interactive traffic*: An example of a character-interactive traffic application is text editing. The main characteristics of this type of application are short frames, low delays, and low throughput.

10.2 FRAME RELAY PROTOCOL ARCHITECTURE

Figure 10.2 depicts the protocol architecture to support the frame-mode bearer service. We need to consider two separate planes of operation: a control (C) plane, which is involved in the establishment and termination of logical connections, and a user (U) plane, which is responsible for the transfer of user data between subscribers. Thus, C-plane protocols are between a subscriber and the network, while U-plane protocols provide end-to-end functionality.

Control Plane

The control plane for frame-mode bearer services is similar to that for common-channel signaling in circuit-switching services, in that a separate logical channel is



* Additional functions to support flow and error control may be provided. LAPF control is one protocol that may be used.

FIGURE 10.2 User-network interface protocol architecture.

used for control information. In the case of ISDN, control signaling is done over the D channel, to control the establishment and termination of frame-mode virtual calls on the D, B, and H channels (see Appendix A).

At the data link layer, LAPD (Q.921) is used to provide a reliable data link control service, with error control and flow control, between user (TE) and network (NT) over the D channel. This data link service is used for the exchange of Q.933 control-signaling messages.

User Plane

For the actual transfer of information between end users, the user-plane protocol is LAPF (Link Access Procedure for Frame-Mode Bearer Services), which is defined in Q.922. Q.922 is an enhanced version of LAPD (Q.921). Only the core functions of LAPF are used for frame relay:

- Frame delimiting, alignment, and transparency
- Frame multiplexing/demultiplexing using the address field
- Inspection of the frame to ensure that it consists of an integral number of octets prior to zero-bit insertion or following zero-bit extraction
- Inspection of the frame to ensure that it is neither too long nor too short
- Detection of transmission errors
- Congestion control functions

The last function listed above is new to LAPF, and is discussed in a later section. The remaining functions listed above are also functions of LAPD.

The core functions of LAPF in the user plane constitute a sublayer of the data link layer; this provides the bare service of transferring data link frames from one subscriber to another, with no flow control or error control. Above this, the user may choose to select additional data link or network-layer end-to-end functions. These are not part of the frame-relay service. Based on the core functions, a network offers frame relaying as a connection-oriented link layer service with the following properties:

- Preservation of the order of frame transfer from one edge of the network to the other
- A small probability of frame loss

Comparison with X.25

As can be seen, this architecture reduces to the bare minimum the amount of work accomplished by the network. User data is transmitted in frames with virtually no processing by the intermediate network nodes, other than to check for errors and to route based on connection number. A frame in error is simply discarded, leaving error recovery to higher layers.

Figure 10.3 compares the protocol architecture of frame-mode bearer service to that of X.25. The packet-handling functions of X.25 operate at layer 3 of the OSI model. At layer 2, LAPB is used. Table 10.1 provides a functional comparison of X.25 and frame relay, and Figure 10.4 illustrates that comparison. As can be seen, the processing burden on the network for X.25 is considerably higher than for frame relay.

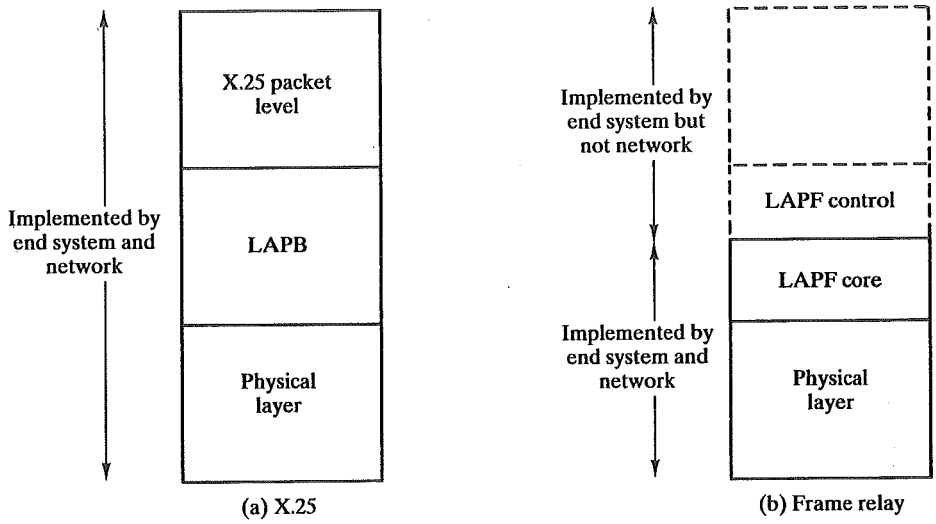


FIGURE 10.3 Comparison of X.25 and frame relay protocol stacks.

TABLE 10.1 Comparison of X.25 packet switching and frame relay.

Function	X.25 in ISDN (X.31)	Frame Relay
Flag generation/recognition	X	X
Transparency	X	X
FCS generation/recognition	X	X
Recognize invalid frames	X	X
Discard incorrect frames	X	X
Address translation	X	X
Fill interframe time	X	X
Multiplexing of logical channels	X	X
Manage V(S) state variable	X	
Manage V(R) state variable	X	
Buffer packets awaiting acknowledgment	X	
Manage retransmission timer T1	X	
Acknowledge received I-frames	X	
Check received N(S) against V(R)	X	
Generation of REJ (rejection message)	X	
Respond to P/F (poll/final) bit	X	
Keep track of number of retransmissions	X	
Act upon reception of REJ	X	
Respond to RNR (receiver not ready)	X	
Respond to RR (receiver ready)	X	
Management of D bit	X	
Management of M bit	X	
Management of Q bit	X	
Management of P(S)	X	
Management of P(R)	X	
Detection of out-of-sequence packets	X	
Management of network layer RR	X	
Management of network layer RNR	X	

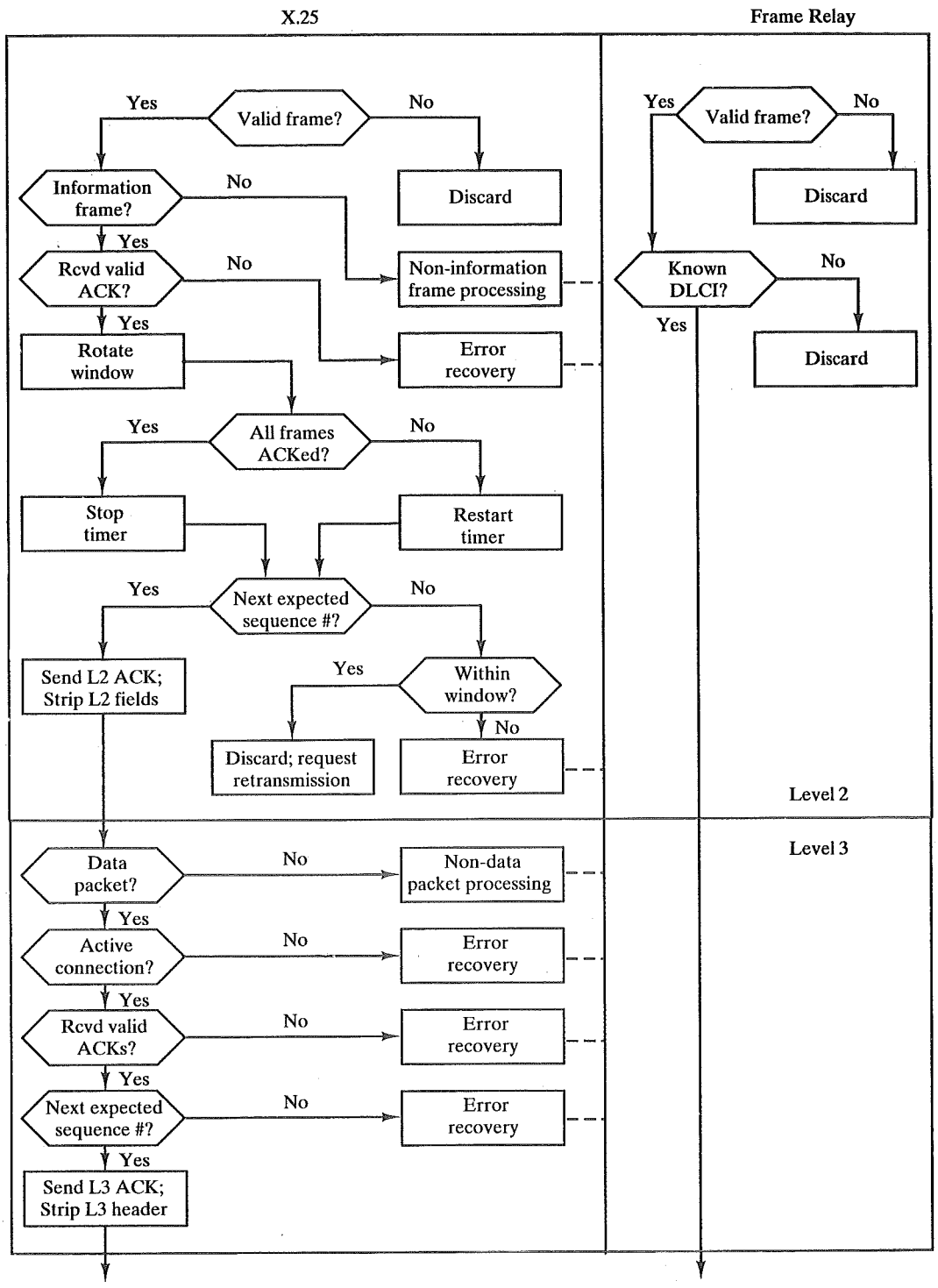


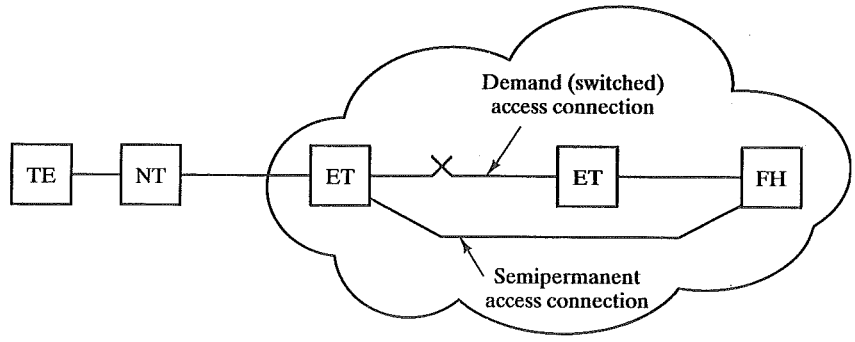
FIGURE 10.4 Simplified model of X.25 and frame relay processing.

10.3 FRAME RELAY CALL CONTROL

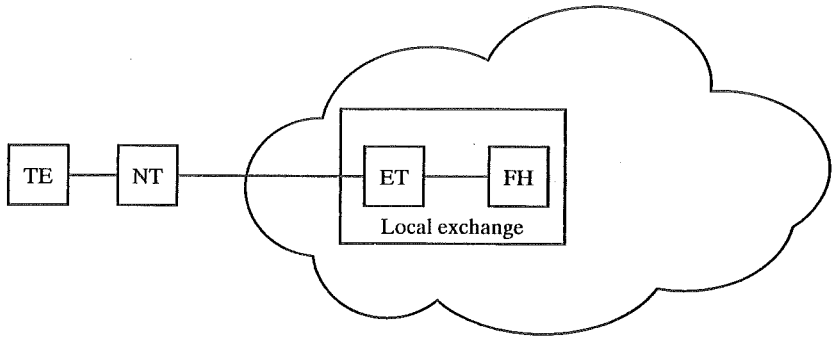
This section examines the various approaches for setting up frame relay connections and then describes the protocol used for connection control.

Call Control Alternatives

The call control protocol for frame relay must deal with a number of alternatives. First, let us consider two cases for the provision of frame handling services. For frame relay operation, a user is not connected directly to another user, but rather to a frame handler in the network; just as for X.25, a user is connected to a packet handler. There are two cases (Figure 10.5):



(a) Switched access



(b) Integrated access

LEGEND
TE = Terminal equipment
NT = Network equipment
ET = Exchange termination
FH = Frame handler

FIGURE 10.5 Frame relay access modes.

- **Switched Access.** The user is connected to a switched network, such as ISDN, and the local exchange does not provide the frame-handling capability. In this case, switched access must be provided from the user's terminal equipment (TE) to the frame handler elsewhere in the network; this can either be a demand connection (set up at the time of the call) or a semi-permanent connection (always available). In either case, the frame relay service is provided over a B or H channel.
- **Integrated Access.** The user is connected to a pure frame-relaying network or to a switched network in which the local exchange does provide the frame-handling capability. In this case, the user has direct logical access to the frame handler.

All of the above considerations have to do with the connection between the subscriber and the frame handler, which we refer to as the *access connection*. Once this connection exists, it is possible to multiplex multiple logical connections, referred to as *frame relay connections*, over this access connection. Such logical connections may be either on-demand or semipermanent.

Frame Relay Connection

The discussion will perhaps be easier to follow if we first consider the management of frame relay connections. So, let us assume that the subscriber has somehow established an access connection to a frame handler that is part of a frame relay network. Analogous to a packet-switching network, the user is now able to exchange data frames with any other user attached to the network. For this purpose, a frame relay connection, analogous to a packet-switching virtual circuit, must first be established between two users.

As with X.25, frame relay supports multiple connections over a single link. In the case of frame relay, these are called data link connections, and each has a unique data link connection identifier (DLCI). Data transfer involves the following stages:

1. Establish a logical connection between two end points, and assign a unique DLCI to the connection.
2. Exchange information in data frames. Each frame includes a DLCI field to identify the connection.
3. Release the logical connection.

The establishment and release of a logical connection is accomplished by the exchange of messages over a logical connection dedicated to call control, with DLCI = 0. A frame with DLCI = 0 contains a call control message in the information field. At a minimum, four message types are needed: SETUP, CONNECT, RELEASE, and RELEASE COMPLETE.

Either side may request the establishment of a logical connection by sending a SETUP message. The other side, upon receiving the SETUP message, must reply with a CONNECT message if it accepts the connection; otherwise, it responds with

a RELEASE COMPLETE message. The side sending the SETUP message may assign the DLCI by choosing an unused value and including this value in the SETUP message; otherwise, the DLCI value is assigned by the accepting side in the CONNECT message.

Either side may request to clear a logical connection by sending a RELEASE message. The other side, upon receipt of this message, must respond with a RELEASE COMPLETE message.

Table 10.2 shows the complete set of call control messages for frame relay. These messages are defined in ITU-T standard Q.933. They are a subset of a larger collection of messages defined in Q.931 used for common-channel signaling between a user and an ISDN.

Access Connection

Now consider the establishment of an access connection. If the connection is semi-permanent, then no call control protocol is required. If the connection is to be set

TABLE 10.2 Messages for frame relay connection control.

Message	Direction	Function
Access connection establishment messages		
ALERTING	u → n	Indicates that user alerting has begun
CALL PROCEEDING	both	Indicates that access connection establishment has been initiated
CONNECT	both	Indicates access connection acceptance by called TE
CONNECT ACKNOWLEDGE	both	Indicates that user has been awarded the access connection
PROGRESS	u → n	Reports progress of an access connection in the event of interworking with a private network
SETUP	both	Initiates access connection establishment
Access connection clearing messages		
DISCONNECT	both	Sent by user to request connection clearing; sent by network to indicate connection clearing
RELEASE	both	Indicates intent to release channel and call reference.
RELEASE COMPLETE	both	Indicates release of channel and call reference
Miscellaneous messages		
STATUS	both	Sent in response to a STATUS ENQUIRY or at any time to report an error
STATUS ENQUIRY	both	Solicits STATUS message

up on demand, then the user requests such a connection by means of a common-channel signaling protocol between the user and the network. In the case of ISDN, and also many other digital networks, the protocol used is Q.931.

Figure 10.6 provides an example of the types of exchanges involved for switched access to a frame handler, in this case over an ISDN. First, the calling user must establish a circuit-switched connection to a frame handler that is one of the nodes of the frame relay network; this is done with the usual SETUP, CONNECT, and CONNECT ACK messages, exchanged at the local user-network interface and at the interface between the network and a frame handler. The procedures and parameters for this exchange are carried out on the D channel, and are defined in Q.931. In the figure, it is assumed that the access connection is created for a B channel.

Once the access connection is established, an exchange takes place directly between the end user and the frame handling node for each frame mode connection that is set up. Again, the SETUP, CONNECT, and CONNECT ACK messages are

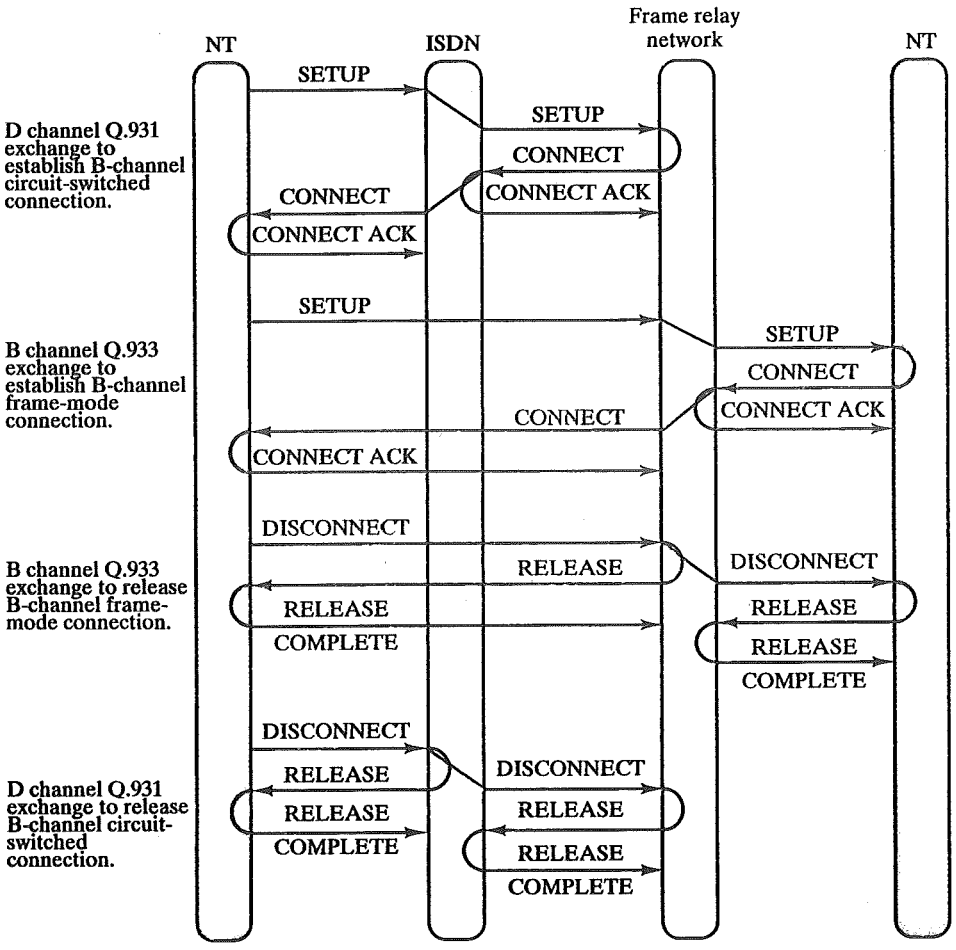


FIGURE 10.6 Example of frame-mode control signaling.

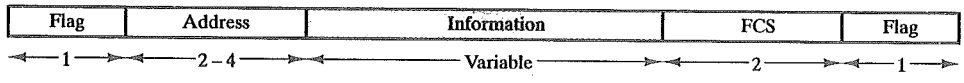
used. In this case, the procedures and parameters for this exchange are defined in Q.933, and the exchange is carried out on the same B channel that will be used for the frame mode connection.

10.4 USER DATA TRANSFER

The operation of frame relay for user data transfer is best explained by beginning with the frame format, illustrated in Figure 10.7a. This is the format defined for the minimum-function LAPF protocol (known as LAPF core protocol). The format is similar to that of LAPD and LAPB with one obvious omission: There is no control field. This has the following implications:

- There is only one frame type, used for carrying user data. There are no control frames.
- It is not possible to use inband signaling; a logical connection can only carry user data.
- It is not possible to perform flow control and error control, as there are no sequence numbers.

The flag and frame check sequence (FCS) fields function as in LAPD and LAPB. The information field carries higher-layer data. If the user selects to implement additional data link control functions end-to-end, then a data link frame can



(a) Frame format

8	7	6	5	4	3	2	1
Upper DLCI						C/R	EA 0
Lower DLCI				FECN	BECN	DE	EA 1

(b) Address field—2 octets (default)

8	7	6	5	4	3	2	1
Upper DLCI						C/R	EA 0
DLCI				FECN	BECN	DE	EA 0
Lower DLCI or DL-Core control						D/C	EA 1

(c) Address field—3 octets

8	7	6	5	4	3	2	1
Upper DLCI						C/R	EA 0
DLCI							0
DLCI				FECN	BECN	DE	EA 0
DLCI							EA 0
Lower DLCI or DL-Core control						D/C	EA 1

(d) Address field—4 octets

LEGEND

EA	Address field extension bit	BECN	Backward explit congestion notification
C/R	Command/response bit	DLCI	Data link congestion identifier
FECN	Forward explicit congestion notification	D/C	DLCI or CORE control indicator

FIGURE 10.7 LAPF-core formats.

be carried in this field. Specifically, a common selection will be to use the full LAPF protocol (known as LAPF control protocol) in order to perform functions above the LAPF core functions. Note that the protocol implemented in this fashion is strictly between the end subscribers and is transparent to ISDN.

The address field has a default length of 2 octets and may be extended to 3 or 4 octets. It carries a data link connection identifier (DLCI) of 10, 17, or 24 bits. The DLCI serves the same function as the virtual circuit number in X.25: It allows multiple logical frame relay connections to be multiplexed over a single channel. As in X.25, the connection identifier has only local significance; each end of the logical connection assigns its own DLCI from the pool of locally unused numbers, and the network must map from one to the other. The alternative, using the same DLCI on both ends, would require some sort of global management of DLCI values.

The length of the address field, and hence of the DLCI, is determined by the address field extension (EA) bits. The C/R bit is application-specific and is not used by the standard frame relay protocol. The remaining bits in the address field have to do with congestion control, and are discussed in Section 10.6.

Figure 10.8 is another view of the protocols involved in frame relay, this time from the point of view of the individual frame relay connections. There is a common physical layer and frame relay sublayer. An optional layer-2 data link control protocol may be included above the frame relay sublayer. This selection is application-dependent and may differ for different frame relay connections (DLC-i). If frame relay call control messages are carried in frame relay frames, they are carried on DLCI 0, which provides a frame relay connection between the user and the frame handler. DLCI 8191 is dedicated to management procedures.

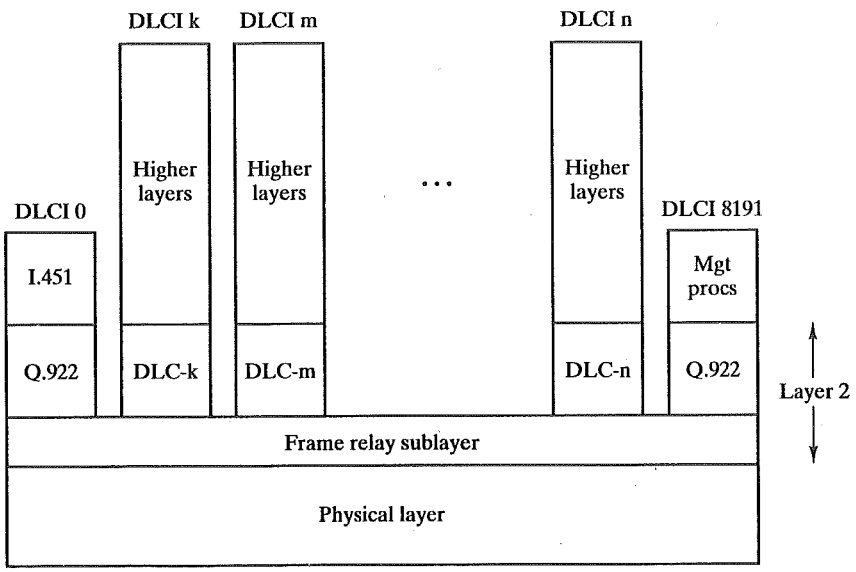


FIGURE 10.8 Multiplexing at the frame relay sublayer.

10.5 NETWORK FUNCTION

The frame relaying function performed by ISDN, or any network that supports frame relaying, consists of the routing of frames with the format of Figure 10.7a, based on their DLCI values.

Figure 10.9 suggests the operation of a frame handler in a situation in which a number of users are directly connected to the same frame handler over different physical channels. The operation could just as well involve relaying a frame through two or more frame handlers. In this figure, the decision-making logic is shown conceptually as a separate module: the frame relay control point. This module is responsible for making routing decisions.

Typically, routing is controlled by entries in a connection table based on DLCI that map incoming frames from one channel to another. The frame handler switches a frame from an incoming channel to an outgoing channel, based on the appropriate entry in the connection table, and translates the DLCI in the frame before transmission. For example, incoming frames from TE B on logical connection 306 are retransmitted to TE D on logical connection 342. The figure also shows the multiplexing function: Multiple logical connections to TE D are multiplexed over the same physical channel.

Note also that all of the TEs have a logical connection to the frame relay control point with a value of DLCI = 0. These connections are reserved for in-channel call control, to be used when I.451/Q.931 on the D-channel is not used for frame relay call control.

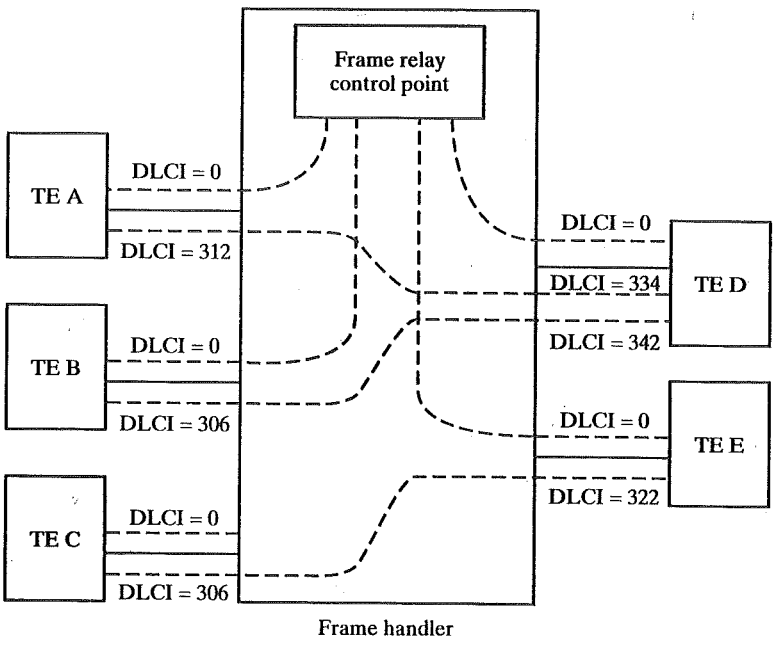


FIGURE 10.9 Frame handler operation.

As part of the frame relay function, the FCS of each incoming frame is checked. When an error is detected, the frame is simply discarded. It is the responsibility of the end users to institute error recovery above the frame relay protocol.

10.6 CONGESTION CONTROL

In Section 9.3, we discussed the potentially disastrous effects of congestion on the ability of network nodes to sustain throughput. To summarize that discussion, Figure 10.10 illustrates the effects of congestion in general terms. As the load on a network increases, a region of mild congestion is reached, where the queuing delays at the nodes results in increased end-to-end delay and reduced capability to provide desired throughput. When a point of severe congestion is reached, the classic queu-

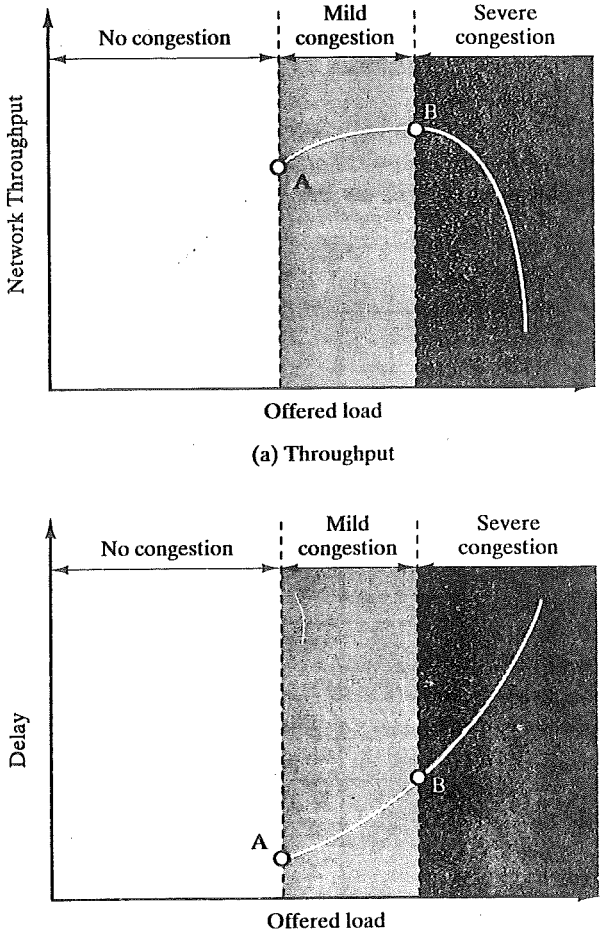


FIGURE 10.10 The effects of congestion.

ing response (see Appendix B) results in dramatic growth in delays and a collapse in throughput.

It is clear that these catastrophic events must be avoided, which is the task of congestion control. The object of all congestion control techniques is to limit queue lengths at the frame handlers so as to avoid throughput collapse. This section provides an overview of congestion control techniques developed as part of the frame relay standardization effort.

Approaches to Frame Relay Congestion Control

ITU-T Recommendation I.370 defines the objectives for frame relay congestion control to be the following:

- Minimize frame discard
- Maintain, with high probability and minimum variance, an agreed quality of service
- Minimize the possibility that one end user can monopolize network resources at the expense of other end users
- Be simple to implement, and place little overhead on either end user or network
- Create minimal additional network traffic
- Distribute network resources fairly among end users
- Limit spread of congestion to other networks and elements within the network
- Operate effectively regardless of the traffic flow in either direction between end users
- Have minimum interaction or impact on other systems in the frame relaying network
- Minimize the variance in quality of service delivered to individual frame relay connections during congestion (e.g., individual logical connections should not experience sudden degradation when congestion approaches or has occurred)

The challenge of congestion control is particularly acute for a frame relay network because of the limited tools available to the frame handlers. The frame relay protocol has been streamlined in order to maximize throughput and efficiency; a consequence of this is that a frame handler cannot control the flow of frames coming from a subscriber or an adjacent frame handler using the typical sliding-window flow control protocol, such as is found in LAPD.

Congestion control is the joint responsibility of the network and the end users. The network (i.e., the collection of frame handlers) is in the best position to monitor the degree of congestion, while the end users are in the best position to control congestion by limiting the flow of traffic.

Table 10.3 lists the congestion control techniques defined in the various ITU-T and ANSI documents. *Discard strategy* deals with the most fundamental response to congestion: When congestion becomes severe enough, the network is forced to discard frames; we would like to do this in a way that is fair to all users.

TABLE 10.3 Frame relay congestion control techniques.

Technique	Type	Function	Key elements
Discard control	Discard strategy	Provide guidance to network concerning which frames to discard.	DE bit
Backward explicit congestion notification	Congestion avoidance	Provides guidance to end systems about congestion in network	BECN bit
Forward explicit congestion notification	Congestion avoidance	Provides guidance to end systems about congestion in network	FECN bit
Implicit congestion notification	Congestion recovery	End system infers congestion from frame loss	Sequence numbers in higher-layer PDU

Congestion avoidance procedures are used at the onset of congestion to minimize the effect on the network. Thus, these procedures would be initiated at or prior to point A in Figure 10.10 to prevent congestion from progressing to point B. Near point A, there would be little evidence available to end users that congestion is increasing. Thus, there must be some *explicit signaling* mechanism from the network that will trigger the congestion avoidance.

Congestion recovery procedures are used to prevent network collapse in the face of severe congestion. These procedures are typically initiated when the network has begun to drop frames due to congestion. Such dropped frames will be reported by some higher layer of software (e.g., LAPF control protocol), and serve as an *implicit signaling* mechanism. Congestion recovery procedures operate around point B and within the region of severe congestion, as shown in Figure 10.10.

ITU-T and ANSI consider congestion avoidance with explicit signaling and congestion recovery with implicit signaling to be complementary forms of congestion control in the frame relaying bearer service.

Traffic Rate Management

As a last resort, a frame relaying network must discard frames to cope with congestion. There is no getting around this fact. Because each frame handler in the network has finite memory available for queuing frames (Figure 9.9), it is possible for a queue to overflow, necessitating the discard of either the most recently arrived frame or some other frame.

The simplest way to cope with congestion is for the frame relaying network to simply discard frames arbitrarily, with no regard to the source of a particular frame. In that case, because there is no reward for restraint, the best strategy for any indi-

vidual end system is to transmit frames as rapidly as possible; this, of course, exacerbates the congestion problem.

To provide for a fairer allocation of resources, the frame relaying bearer service includes the concept of a committed information rate (CIR). This is a rate, in bits per second, that the network agrees to support for a particular frame-mode connection. Any data transmitted in excess of the CIR is vulnerable to discard in the event of congestion. Despite the use of the term *committed*, there is no guarantee that even the CIR will be met. In cases of extreme congestion, the network may be forced to provide a service at less than the CIR for a given connection. However, when it comes time to discard frames, the network will choose to discard frames on connections that are exceeding their CIR before discarding frames that are within their CIR.

In theory, each frame relaying node should manage its affairs so that the aggregate of CIRs of all the connections of all the end systems attached to the node do not exceed the capacity of the node. In addition, the aggregate of the CIRs should not exceed the physical data rate across the user-network interface, known as the access rate. The limitation imposed by the access rate can be expressed as follows:

$$\sum_i \text{CIR}_{i,j} \leq \text{Access Rate}_j \tag{10.1}$$

where

$\text{CIR}_{i,j}$ = Committed information rate for connection i on channel j
AccessRate_i = Data rate of user access channel i (D, B, or H)

Considerations of node capacity may result in the selection of lower values for some of the CIRs.

For permanent frame relay connections, the CIR for each connection must be established at the time the connection is made between user and network. For switched connections, the CIR parameter is negotiated; this is done in the setup phase of the call control protocol.

The CIR provides a way of discriminating among frames in determining which frames to discard in the face of congestion. Discrimination is indicated by means of the discard eligibility (DE) bit in the LAPF frame (Figure 10.7). The frame handler to which the user's station attaches performs a metering function (Figure 10.11). If the user is sending data at less than the CIR, the incoming frame handler does not alter the DE bit. If the rate exceeds the CIR, the incoming frame handler will set the DE bit on the excess frames and then forward them; such frames may get through or may be discarded if congestion is encountered. Finally, a maximum rate is defined, such that any frames above the maximum are discarded at the entry frame handler.

The CIR, by itself, does not provide much flexibility in dealing with traffic rates. In practice, a frame handler measures traffic over each logical connection for a time interval specific to that connection, and then makes a decision based on the amount of data received during that interval. Two additional parameters, assigned

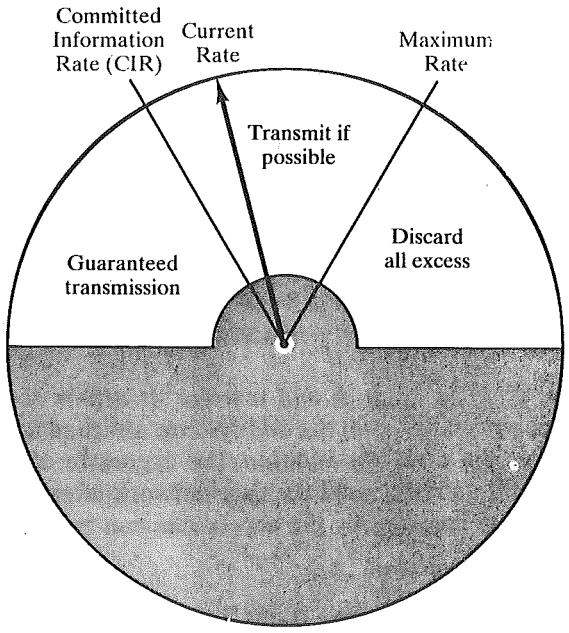


FIGURE 10.11 Operation of the CIR.

on permanent connections and negotiated on switched connections, are needed. They are

- **Committed Burst Size (B_c).** The maximum amount of data that the network agrees to transfer, under normal conditions, over a measurement interval T. These data may or may not be contiguous (i.e., it may appear in one frame or in several frames).
- **Excess Burst Size (B_e).** The maximum amount of data in excess of B_c that the network will attempt to transfer, under normal conditions, over a measurement interval T. These data are uncommitted in the sense that the network does not commit to delivery under normal conditions. Put another way, the data that represent B_e are delivered with lower probability than the data within B_c.

The quantities B_c and CIR are related. Because B_c is the amount of committed data that may be transmitted by the user over a time T, and CIR is the rate at which committed data may be transmitted, we must have

$$T = \frac{B_c}{\text{CIR}} \tag{10.2}$$

Figure 10.12, based on one in ITU-T I.370, illustrates the relationship among these parameters. On each graph, the solid line plots the cumulative number of information bits transferred over a given connection since time T_0. The dashed line labeled Access Rate represents the data rate over the channel containing this con-

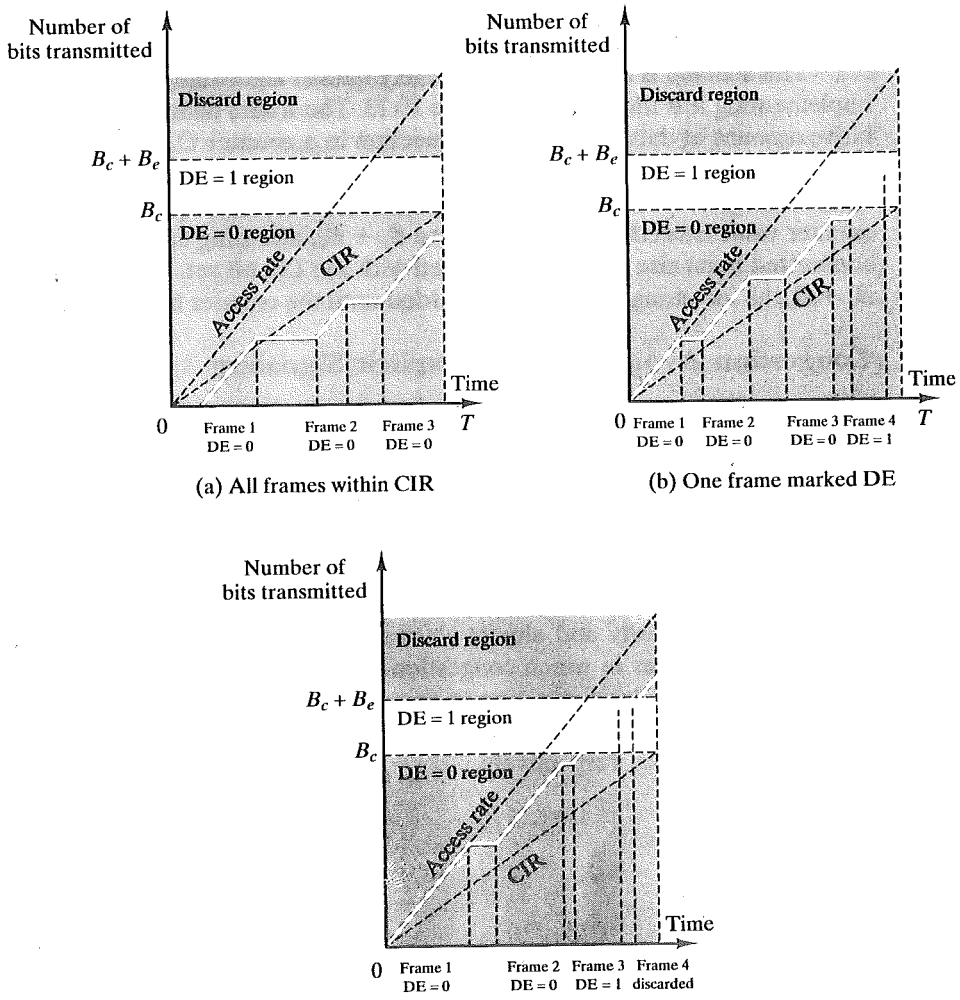


FIGURE 10.12 Illustration of relationships among congestion parameters.

nection. The dashed line labeled CIR represents the committed information rate over the measurement interval T. Note that when a frame is being transmitted, the solid line is parallel to the Access Rate line; when a frame is transmitted on a channel, that channel is dedicated to the transmission of that frame. When no frame is being transmitted, the solid line is horizontal.

Part (a) of the figure shows an example in which three frames are transmitted within the measurement interval, and the total number of bits in the three frames is less than B_c. Note that during the transmission of the first frame, the actual transmission rate temporarily exceeds the CIR. This excess is of no consequence because the frame handler is only concerned with the cumulative number of bits transmitted over the entire interval. In part (b) of the figure, the last frame transmitted during the interval causes the cumulative number of transmitted bits to exceed B_c. Accordingly, that DE bit of that frame is set by the frame handler. In part (c) of the figure,

the third frame exceeds B_c and so is labeled for potential discard. The fourth frame exceeds $B_c + B_e$ and is discarded.

This scheme is an example of a leaky bucket algorithm, and a mechanism for implementing it is illustrated in Figure 10.13. The frame handler records the cumulative amount of data sent over a connection in a counter C. The counter is decremented at a rate of B_c every T time units. Of course, the counter is not allowed to become negative, so the actual assignment is $C \leftarrow \text{MIN}\ [C, B_c]$. Whenever the counter value exceeds B_c but is less than $B_c + B_e$, incoming data are in excess of the committed burst size and are forwarded with the DE bit set. If the counter reaches $B_c + B_e$, all incoming frames are discarded until the counter has been decremented.

Congestion Avoidance with Explicit Signaling

It is desirable to use as much of the available capacity in a frame relay network as possible but still react to congestion in a controlled and fair manner. This is the purpose of explicit congestion avoidance techniques. In general terms, for explicit congestion avoidance, the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network.

As the standards for explicit congestion avoidance were being developed, two general strategies were considered [BERG91]. One group believed that congestion always occurred slowly and almost always in the network egress nodes. Another group had seen cases in which congestion grew very quickly in the internal nodes and required quick, decisive action to prevent network congestion. We will see that these two approaches are reflected in the forward and backward explicit congestion avoidance techniques, respectively.

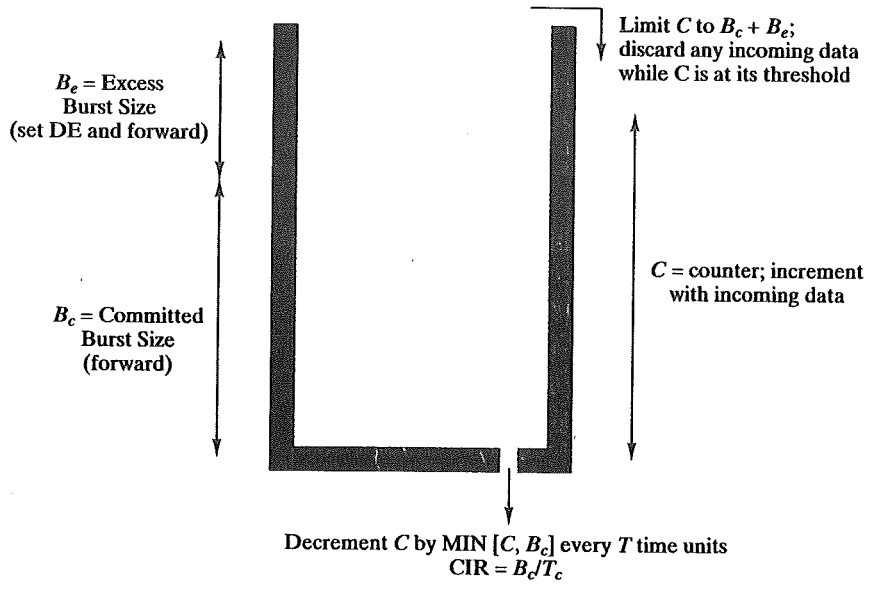


FIGURE 10.13 Leaky bucket algorithm.

For explicit signaling, two bits in the address field of each frame are provided. Either bit may be set by any frame handler that detects congestion. If a frame handler receives a frame in which one or both of these bits are set, it must not clear the bits before forwarding the frame. Thus, the bits constitute signals from the network to the end user. The two bits are

- **Backward explicit congestion notification (BECN).** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received frame. The notification indicates that the frames transmitted by the user on this logical connection may encounter congested resources.
- **Forward explicit congestion notification (FECN).** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the same direction as the received frame. The notification indicates that this frame, on this logical connection, has encountered congested resources.

Let us consider how these bits are used by the network and the user. First, for the *network response*, it is necessary for each frame handler to monitor its queuing behavior. If queue lengths begin to grow to a dangerous level, then either FECN or BECN bits, or a combination, should be set to try to reduce the flow of frames through that frame handler. The choice of FECN or BECN may be determined by whether the end users on a given logical connection are prepared to respond to one or the other of these bits; this may be determined at configuration time. In any case, the frame handler has some choice as to which logical connections should be alerted to congestion. If congestion is becoming quite serious, all logical connections through a frame handler might be notified. In the early stages of congestion, the frame handler might just notify users for those connections that are generating the most traffic.

In an appendix to ANSI T1.618, a procedure for monitoring queue lengths is suggested. The frame handler monitors the size of each of its queues. A cycle begins when the outgoing circuit goes from idle (queue empty) to busy (non-zero queue size, including the current frame). The average queue size over the previous cycle and the current cycle is calculated. If the average size exceeds a threshold value, then the circuit is in a state of incipient congestion, and the congestion avoidance bits should be set on some or all logical connections that use that circuit. By averaging over two cycles instead of just monitoring current queue length, the system avoids reacting to temporary surges that would not necessarily produce congestion.

The average queue length may be computed by determining the area (product of queue size and time interval) over the two cycles and dividing by the time of the two cycles. This algorithm is illustrated in Figure 10.14.

The *user response* is determined by the receipt of BECN or FECN signals. The simplest procedure is the response to a BECN signal: The user simply reduces the rate at which frames are transmitted until the signal ceases. The response to an FECN is more complex, as it requires the user to notify its peer user of this connection to restrict its flow of frames. The core functions used in the frame relay protocol do not support this notification; therefore, it must be done at a higher layer,

t = current time
t_i = time of i^{th} arrival or departure event
f_{qi} = number of frames in the system after the event
T_0 = time at the beginning of the previous cycle
T_1 = time at the beginning of the current cycle

The algorithm consists of three components:

1. Queue Length Update: Beginning with $q_0 := 0$
   If the i^{th} event is an arrival event, $q_i := q_{i-1} + 1$
   If the i^{th} event is a departure event, $q_i := q_{i-1} - 1$
2. Queue Area (integral) update:

   $$\text{Area of the previous cycle} = \sum_{t_i \in [T_0, T_1)} q_{i-1}(t_i - t_{i-1})$$

   $$\text{Area of the current cycle} = \sum_{t_i \in [T_1, t)} q_{i-1}(t_i - t_{i-1})$$

3. Average Queue Length Update
   Average queue length over the two cycles

   $$= \frac{\text{Area of the two cycles}}{\text{Time of the two cycles}} = \frac{\text{Area of the two cycles}}{t - T_0}$$

FIGURE 10.14 Queue length averaging algorithm.

such as the transport layer. The flow control could also be accomplished by the LAPF control protocol or some other link control protocol implemented above the frame relay sublayer (Figure 10.8). The LAPF control protocol is particularly useful because it includes an enhancement to LAPD that permits the user to adjust window size.

Congestion Recovery with Implicit Signaling

Implicit signaling occurs when the network discards a frame, and this fact is detected by the end user at a higher, end-to-end layer, such as the LAPF control protocol. When this occurs, the end user software may deduce that congestion exists.

For example, in a data link control protocol such as the LAPF control protocol, which uses a sliding-window flow and error control technique, the protocol detects the loss of an I frame in one of two ways:

1. When a frame is dropped by the network, the following frame will generate an REJ frame from the receiving end point.
2. When a frame is dropped by the network, no acknowledgment is returned from the other end system. Eventually, the source end system will time out and transmit a command with the P bit set to 1. The subsequent response with the F bit set to 1 should indicate that the receive sequence number N(R) from the other side is less than the current send sequence number.

Once congestion is detected, the protocol uses flow control to recover from the congestion. LAPF suggests that a user that is capable of varying the flow control window size use this mechanism in response to implicit signaling. Let us assume

that the layer-2 window size, W, can vary between the parameters W_{min} and W_{max}, and is initially set to W_{max}. In general, we would like to reduce W, as congestion increases, to gradually throttle the transmission of frames. Three classes of adaptive window schemes based on response to one of the two conditions listed above have been suggested [CHEN89, DOSH88]:

1.1 Set $W = \text{Max}\,[W - 1, W_{min}]$
1.2 Set $W = W_{min}$
1.3 Set $W = \text{Max}\,[\alpha W, W_{min}]$, where $0 < \alpha < 1$

Successful transmissions (measured by receipt of acknowledgments) may indicate that the congestion has gone away and window size should be increased. Two possible approaches are

2.1 Set $W = \text{Min}\,[W + 1, W_{max}]$ after N consecutive successful transmissions
2.2 Set $W = \text{Min}\,[W + 1, W_{max}]$ after W consecutive successful transmissions

A study reported in [CHEN89] suggests that the use of strategy 1.3 with $\alpha = 0.5$ plus strategy 2.2 provides good performance over a wide range of network parameters and traffic patterns; this is the strategy recommended in LAPF.

10.7 RECOMMENDED READING

[SMIT93] provides a good survey of frame relay, with an emphasis on its applications and its role in the context of other networking services; the study also provides an overview of the specifications. [BLAC94] provides a greater emphasis on the technical and protocol aspects of frame relay. Another good technical treatment is contained in [SPOH93].

BLAC94 Black, U. *Frame Relay Networks: Specifications and Implementations*. New York: McGraw-Hill, 1994.

SMIT93 Smith, P. *Frame Relay: Principles and Applications*. Reading, MA: Addison-Wesley, 1993.

SPOH93 Spohn, D. *Data Network Design*. New York: McGraw-Hill, 1994.

Recommended Web Sites

- http://www.frforum.com: Web site of the Frame Relay Forum, which is leading the effort to expand the functionality of frame relay networks.
- http://www.mot.com/MIMS/ISG/tech/frame-relay/resources.html: Exhaustive source of information on frame relay.

10.8 PROBLEMS

10.1 A proposed congestion control technique is known as isarithmic control. In this method, the total number of frames in transit is fixed by inserting a fixed number of permits into the network. These permits circulate at random through the frame relay

network. Whenever a frame handler wants to relay a frame just given to it by an attached user, it must first capture and destroy a permit. When the frame is delivered to the destination user by the frame handler to which it attaches, that frame handler reissues the permit. List three potential problems with this technique.

10.2 Consider the frame relay network depicted in Figure 10.15. C is the capacity of a link in frames per second. Node A presents a constant load of 0.8 frames per second destined for A'. Node B presents a load λ destined for B'. Node S has a common pool of buffers that it uses for traffic both to A' and B'. When the buffer is full, frames are discarded and are later retransmitted by the source user. S has a throughput capacity of 2. Plot the total throughput (i.e., the sum of A–A' and B–B' delivered traffic) as a function of λ. What fraction of the throughput is A–A' traffic for $\lambda > 1$?

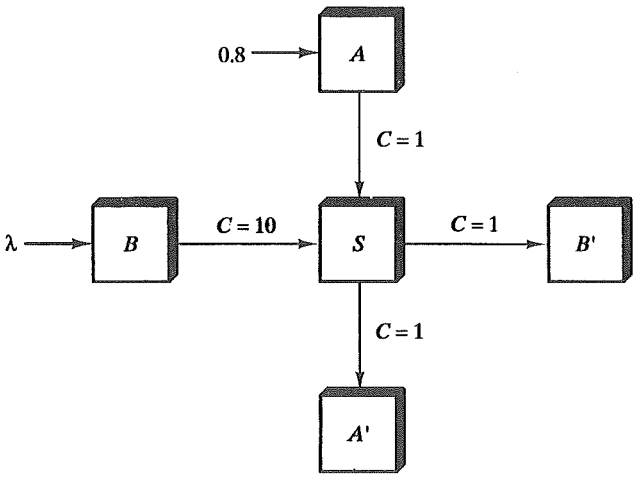
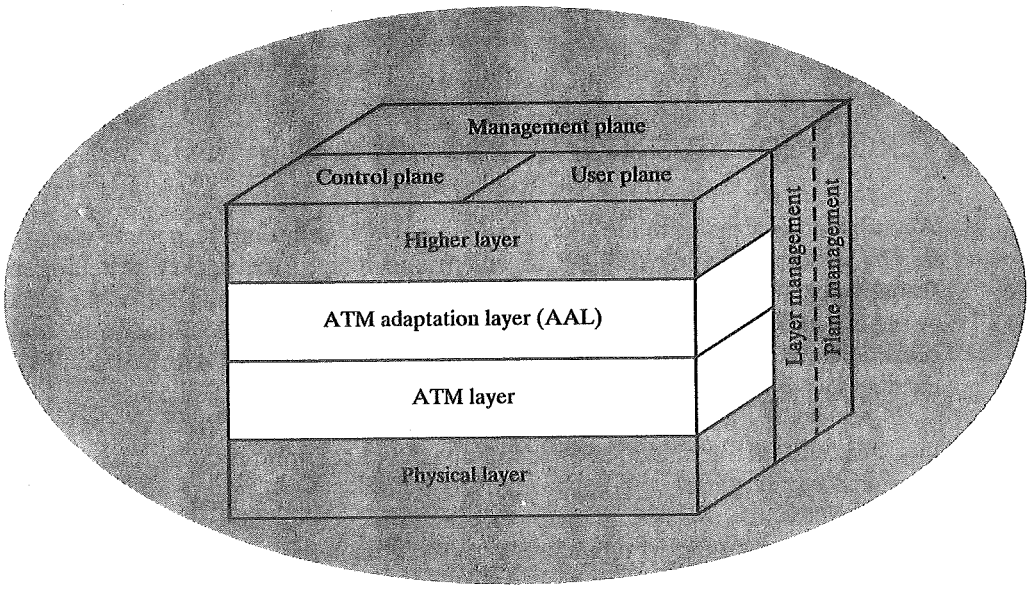


FIGURE 10.15 Network of nodes.

CHAPTER 11

ASYNCHRONOUS TRANSFER MODE (ATM)



Asynchronous transfer mode (ATM), also known as cell relay, is similar in concept to frame relay. Both frame relay and ATM take advantage of the reliability and fidelity of modern digital facilities to provide faster packet switching than X.25. ATM is even more streamlined than frame relay in its functionality, and can support data rates several orders of magnitude greater than frame relay.

In addition to their technical similarities, ATM and frame relay have similar histories. Frame relay was developed as part of the work of ISDN, but is now finding wide application in private networks and other non-ISDN applications, particularly in bridges and routers. ATM was developed as part of the work on broadband ISDN, but is beginning to find application in non-ISDN environments where very high data rates are required.

We begin with a discussion of the details of the ATM scheme. Then, the important concept of the ATM Adaptation Layer (AAL) is examined. Finally, the key issue of congestion control in ATM networks is discussed. For a discussion of broadband ISDN, see Appendix A.

11.1 PROTOCOL ARCHITECTURE

Asynchronous transfer mode (ATM), also known as cell relay, is in some ways similar to packet switching using X.25 and frame relay. Like packet switching and frame relay, ATM involves the transfer of data in discrete chunks. Also, like packet switching and frame relay, ATM allows multiple logical connections to be multiplexed over a single physical interface. In the case of ATM, the information flow on each logical connection is organized into fixed-size packets, called cells.

ATM is a streamlined protocol with minimal error and flow control capabilities; this reduces the overhead of processing ATM cells and reduces the number of overhead bits required with each cell, thus enabling ATM to operate at high data rates. Further, the use of fixed-size cells simplifies the processing required at each ATM node, again supporting the use of ATM at high data rates.

The standards issued for ATM by ITU-T are based on the protocol architecture shown in Figure 11.1, which illustrates the basic architecture for an interface between user and network. The physical layer involves the specification of a transmission medium and a signal encoding scheme. The data rates specified at the physical layer include 155.52 Mbps and 622.08 Mbps. Other data rates, both higher and lower, are possible.

Two layers of the protocol architecture relate to ATM functions. There is an ATM layer common to all services that provides packet transfer capabilities, and an ATM adaptation layer (AAL) that is service dependent. The ATM layer defines the transmission of data in fixed-size cells and also defines the use of logical connections. The use of ATM creates the need for an adaptation layer to support information transfer protocols not based on ATM. The AAL maps higher-layer information into ATM cells to be transported over an ATM network, then collects information from ATM cells for delivery to higher layers.

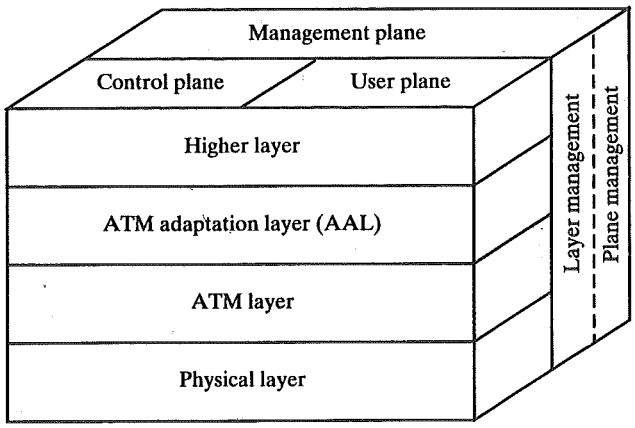


FIGURE 11.1 ATM protocol reference model.

The protocol reference model makes reference to three separate planes:

- **User Plane.** Provides for user information transfer, along with associated controls (e.g., flow control, error control).
- **Control Plane.** Performs call control and connection control functions.
- **Management Plane.** Includes plane management, which performs management functions related to a system as a whole and provides coordination between all the planes, and layer management, which performs management functions relating to resources and parameters residing in its protocol entities.

11.2 ATM LOGICAL CONNECTIONS

Logical connections in ATM are referred to as virtual channel connections (VCC). A VCC is analogous to a virtual circuit in X.25 or a data link connection in frame relay; it is the basic unit of switching in an ATM network. A VCC is set up between two end users through the network and a variable-rate, full-duplex flow of fixed-size cells is exchanged over the connection. VCCs are also used for user-network exchange (control signaling) and network-network exchange (network management and routing).

For ATM, a second sublayer of processing has been introduced that deals with the concept of virtual path (Figure 11.2). A virtual path connection (VPC) is a bundle of VCCs that have the same endpoints. Thus, all of the cells flowing over all of the VCCs in a single VPC are switched together.

The virtual path concept was developed in response to a trend in high-speed networking in which the control cost of the network is becoming an increasingly higher proportion of the overall network cost. The virtual-path technique helps contain the control cost by grouping connections that share common paths through the network into a single unit. Network management actions can then be applied to a

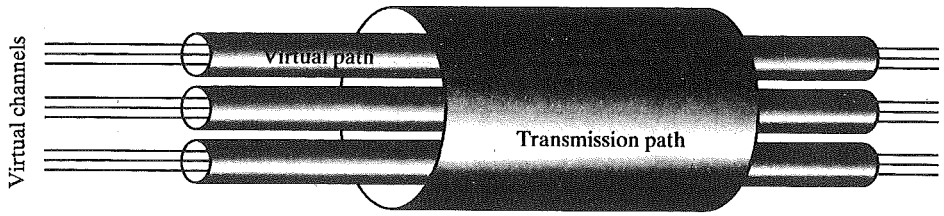


FIGURE 11.2 ATM connection relationships.

small number of groups of connections instead of to a large number of individual connections.

Several advantages can be listed for the use of virtual paths:

- **Simplified network architecture.** Network transport functions can be separated into those related to an individual logical connection (virtual channel) and those related to a group of logical connections (virtual path).
- **Increased network performance and reliability.** The network deals with fewer, aggregated entities.
- **Reduced processing and short connection setup time.** Much of the work is done when the virtual path is set up. By reserving capacity on a virtual path connection in anticipation of later call arrivals, new virtual channel connections can be established by executing simple control functions at the endpoints of the virtual path connection; no call processing is required at transit nodes. Thus, the addition of new virtual channels to an existing virtual path involves minimal processing.
- **Enhanced network services.** The virtual path is used internal to the network but is also visible to the end user. As a result, the user may define closed user groups or closed networks of virtual-channel bundles.

Figure 11.3 suggests in a general way the call-establishment process using virtual channels and virtual paths. The process of setting up a virtual path connection is decoupled from the process of setting up an individual virtual channel connection:

- The virtual path control mechanisms include calculating routes, allocating capacity, and storing connection state information.
- To set up a virtual channel, there must first be a virtual path connection to the required destination node with sufficient available capacity to support the virtual channel, with the appropriate quality of service. A virtual channel is set up by storing the required state information (virtual channel/virtual path mapping).

The terminology of virtual paths and virtual channels used in the standard is a bit confusing, and is summarized in Table 11.1. Whereas most of the network-layer protocols that we survey in this book relate only to the user-network interface, the concepts of virtual path and virtual channel are defined in the ITU-T Recom-

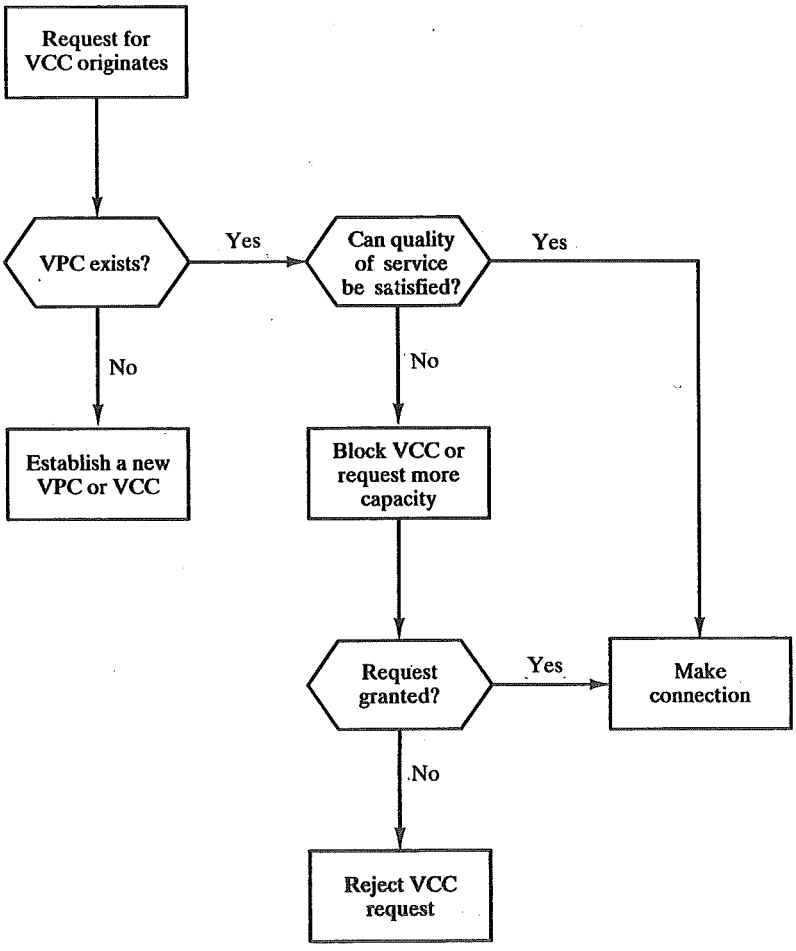


FIGURE 11.3 Call establishment using virtual paths.

mendations with reference to both the user-network interface and the internal network operation.

Virtual Channel Connection Uses

The endpoints of a VCC may be end users, network entities, or an end user and a network entity. In all cases, cell sequence integrity is preserved within a VCC; that is, cells are delivered in the same order in which they are sent. Let us consider examples of the three uses of a VCC:

- **Between end users.** Can be used to carry end-to-end user data; can also be used to carry control signaling between end users, as explained below. A VPC between end users provides them with an overall capacity; the VCC organization of the VPC is up to the two end users, provided the set of VCCs does not exceed the VPC capacity.

TABLE 11.1 Virtual path/virtual connection terminology.

Virtual Channel (VC)	A generic term used to describe unidirectional transport of ATM cells associated by a common unique identifier value.
Virtual Channel Link	A means of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that value is translated or terminated.
Virtual Channel Identifier (VCI)	Identifies a particular VC link for a given VPC.
Virtual Channel Connection (VCC)	A concatenation of VC links that extends between two points where the adaptation layer is accessed. VCCs are provided for the purpose of user-user, user-network, or network-network information transfer. Cell sequence integrity is preserved for cells belonging to the same VCC.
Virtual Path	A generic term used to describe unidirectional transport of ATM cells belonging to virtual channels that are associated by a common unique identifier value.
Virtual Path Link	A group of VC links, identified by a common value of VPI, between a point where a VPI value is assigned and the point where that value is translated or terminated.
Virtual Path Identifier (VPI)	Identifies a particular VP link.
Virtual Path Connection (VPC)	A concatenation of VP links that extends between the point where the VCI values are assigned and the point where those values are translated or removed, i.e., extending the length of a bundle of VC links that share the same VPI. VPCs are provided for the purpose of user-user, user-network, or network-network information transfer.

- **Between an end user and a network entity.** Used for user-to-network control signaling, as discussed below. A user-to-network VPC can be used to aggregate traffic from an end user to a network exchange or network server.
- **Between two network entities.** Used for network traffic management and routing functions. A network-to-network VPC can be used to define a common route for the exchange of network management information.

Virtual Path/Virtual Channel Characteristics

ITU-T Recommendation I.150 lists the following as characteristics of virtual channel connections:

- **Quality of service.** A user of a VCC is provided with a Quality of Service specified by parameters such as cell loss ratio (ratio of cells lost to cells transmitted) and cell delay variation.
- **Switched and semi-permanent virtual channel connections.** Both are switched connections, which require call-control signaling, and dedicated channels can be provided.
- **Cell sequence integrity.** The sequence of transmitted cells within a VCC is preserved.
- **Traffic parameter negotiation and usage monitoring.** Traffic parameters can be negotiated between a user and the network for each VCC. The input of

cells to the VCC is monitored by the network to ensure that the negotiated parameters are not violated.

The types of traffic parameters that can be negotiated include average rate, peak rate, burstiness, and peak duration. The network may need a number of strategies to handle congestion and to manage existing and requested VCCs. At the crudest level, the network may simply deny new requests for VCCs to prevent congestion. Additionally, cells may be discarded if negotiated parameters are violated or if congestion becomes severe. In an extreme situation, existing connections might be terminated.

I.150 also lists characteristics of VPCs. The first four characteristics listed are identical to those for VCCs. That is, quality of service, switched and semi-permanent VPCs, cell sequence integrity, and traffic parameter negotiation and usage monitoring are all also characteristics of a VPC. There are a number of reasons for this duplication. First, redundancy provides some flexibility in how the network service manages its requirements. Second, the network must be concerned with the overall requirements for a VPC, and, within a VPC, it may negotiate the establishment of virtual channels with given characteristics. Finally, once a VPC is set up, it is possible for the end users to negotiate the creation of new VCCs. The VPC characteristics impose a discipline on the choices that the end users may make.

In addition, a fifth characteristic is listed for VPCs:

- **Virtual channel identifier restriction within a VPC.** One or more virtual channel identifiers, or numbers, may not be available to the user of the VPC, but may be reserved for network use. Examples include VCCs used for network management.

Control Signaling

In ATM, a mechanism is needed for the establishment and release of VPCs and VCCs. The exchange of information involved in this process is referred to as control signaling, and takes place on separate connections from those that are being managed.

For VCCs, I.150 specifies four methods for providing an establishment/release facility. One or a combination of these methods will be used in any particular network:

1. *Semi-permanent VCCs* may be used for user-to-user exchange. In this case, no control signaling is required.
2. If there is no pre-established call control signaling channel, then one must be set up. For that purpose, a control signaling exchange must take place between the user and the network on some channel. Hence, we need a permanent channel, probably of low data rate, that can be used to set up VCCs that can be used for call control. Such a channel is called a *meta-signaling channel*, as the channel is used to set up signaling channels.
3. The meta-signaling channel can be used to set up a VCC between the user and the network for call control signaling. This *user-to-network signaling virtual channel* can then be used to set up VCCs to carry user data.

4. The meta-signaling channel can also be used to set up a *user-to-user signaling virtual channel.* Such a channel must be set up within a pre-established VPC. It can then be used to allow the two end users, without network intervention, to establish and release user-to-user VCCs to carry user data.

For VPCs, three methods are defined in I.150:

1. A VPC can be established on a *semi-permanent* basis by prior agreement. In this case, no control signaling is required.
2. VPC establishment/release may be *customer controlled.* In this case, the customer uses a signaling VCC to request the VPC from the network.
3. VPC establishment/release may be *network controlled.* In this case, the network establishes a VPC for its own convenience. The path may be network-to-network, user-to-network, or user-to-user.

11.3 ATM CELLS

The asynchronous transfer mode makes use of fixed-size cells, which consist of a 5-octet header and a 48-octet information field. There are several advantages to the use of small, fixed-size cells. First, their use may reduce queuing delay for a high-priority cell, as it waits less if it arrives slightly behind a lower-priority cell that has gained access to a resource (e.g., the transmitter). Secondly, it appears that fixed-size cells can be switched more efficiently, which is important for the very high data rates of ATM. With fixed-size cells, it is easier to implement the switching mechanism in hardware.

Header Format

Figure 11.4a shows the header format at the user-network interface. Figure 11.4b shows the cell header format internal to the network, where the generic flow control field, which performs end-to-end functions, is not retained. Instead, the virtual path identifier field is expanded from 8 to 12 bits; this allows support for an expanded number of VPCs internal to the network, to include those supporting subscribers and those required for network management.

The *generic flow control field* does not appear in the cell header internal to the network, but only at the user-network interface. Hence, it can be used for control of cell flow only at the local user-network interface. The details of its application are for further study. The field could be used to assist the customer in controlling the flow of traffic for different qualities of service. One candidate for the use of this field is a multiple-priority level indicator to control the flow of information in a service-dependent manner. In any case, the GFC mechanism is used to alleviate short-term overload conditions in the network.

The *virtual path identifier* (VPI) constitutes a routing field for the network. It is 8 bits at the user-network interface and 12 bits at the network-network interface, allowing for more virtual paths to be supported within the network. The *virtual*

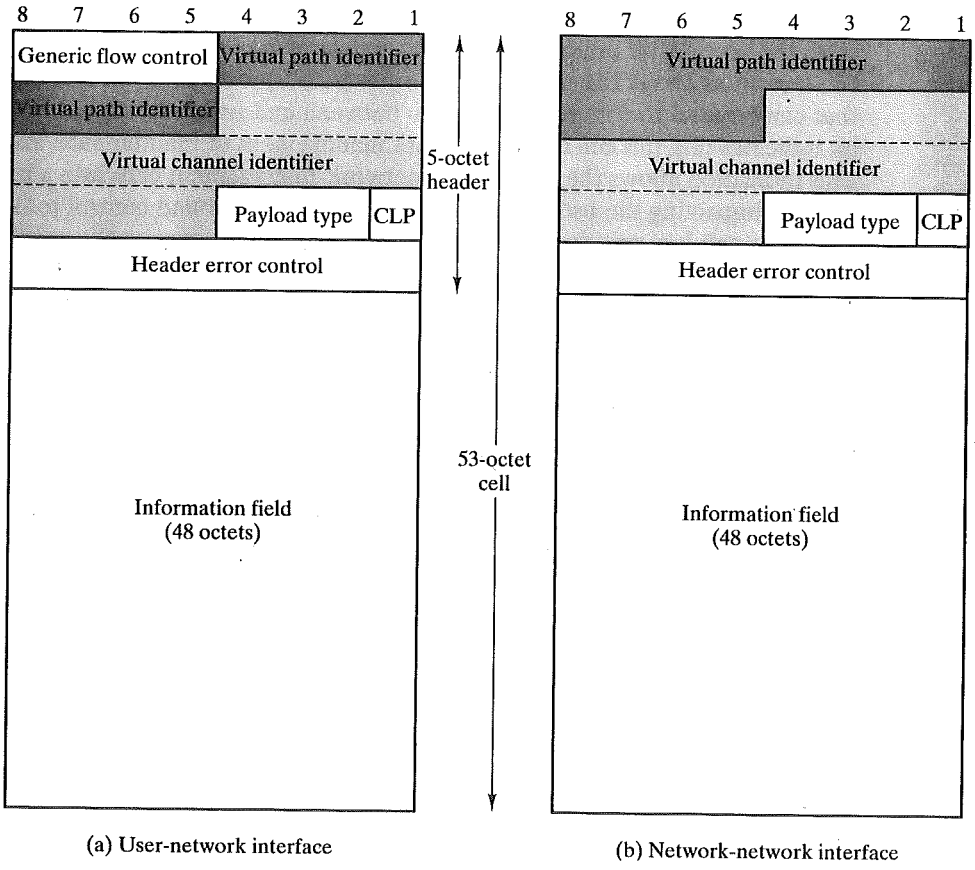


FIGURE 11.4 ATM cell format.

channel identifier (VCI) is used for routing to and from the end user. Thus, it functions much as a service access point.

The *payload-type* field indicates the type of information in the information field. Table 11.2 shows the interpretation of the PT bits. A value of 0 in the first bit

TABLE 11.2 Payload Type (PT) field coding.

PT coding	Interpretation
0 0 0	User data cell, AAU = 0, congestion not experienced
0 0 1	User data cell, AAU = 1, congestion not experienced
0 1 0	User data cell, AAU = 0, congestion experienced
0 1 1	User data cell, AAU = 1, congestion experienced
1 0 0	OAM F5 segment associated cell
1 0 1	OAM F5 end-to-end associated cell
1 1 0	Resource management cell
1 1 1	Reserved for future function

AAU = ATM user to ATM user indication

indicates user information—that is, information from the next higher layer. In this case, the second bit indicates whether congestion has been experienced; the third bit, known as the ATM-user-to-ATM-user (AAU) indication bit is a one-bit field that can be used to convey information between end users. A value of 1 in the first bit indicates that this cell carries network management or maintenance information. This indication allows the insertion of network-management cells into a user's VCC without impacting the user's data, thereby providing in-band control information.

The **cell-loss priority** (CLP) is used to provide guidance to the network in the event of congestion. A value of 0 indicates a cell of relatively higher priority, which should be discarded only when no other alternative is available. A value of 1 indicates that this cell is subject to discard within the network. The user might employ this field so that extra information may be inserted into the network, with a CLP of 1, and delivered to the destination if the network is not congested. The network may set this field to 1 for any data cell that is in violation of an agreement concerning traffic parameters between the user and the network. In this case, the switch that does the setting realizes that the cell exceeds the agreed traffic parameters but that the switch is capable of handling the cell. At a later point in the network, if congestion is encountered, this cell has been marked for discard in preference to cells that fall within agreed traffic limits.

Header Error Control

Each ATM cell includes an 8-bit header error control field (HEC) that is calculated based on the remaining 32 bits of the header. The polynomial used to generate the code is $X^8 + X^2 + X + 1$. In most existing protocols that include an error control field, such as HDLC and LAPF, the data that serve as input to the error code calculation are, in general, much longer than the size of the resulting error code; this allows for error detection. In the case of ATM, the input to the calculation is only 32 bits, compared to 8 bits for the code. The fact that the input is relatively short allows the code to be used not only for error detection but, in some cases, for actual error correction; this is because there is sufficient redundancy in the code to recover from certain error patterns.

Figure 11.5 depicts the operation of the HEC algorithm at the receiver. At initialization, the receiver's error-correction algorithm is in the default mode for single-

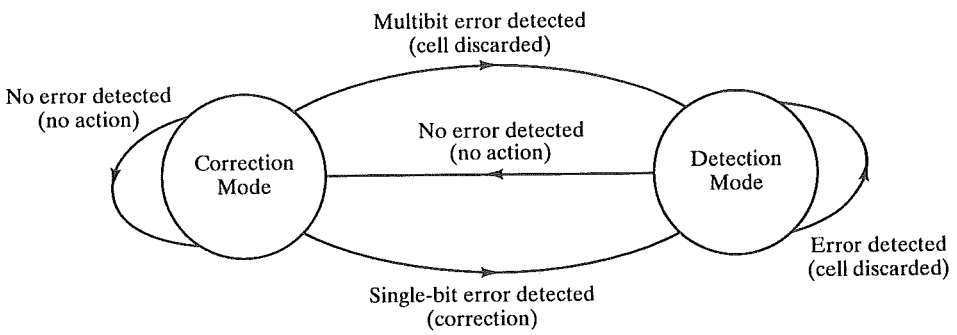


FIGURE 11.5 HEC operation at receiver.

bit error correction. As each cell is received, the HEC calculation and comparison is performed. As long as no errors are detected, the receiver remains in error-correction mode. When an error is detected, the receiver will correct the error if it is a single-bit error or it will detect that a multi-bit error has occurred. In either case, the receiver now moves to detection mode. In this mode, no attempt is made to correct errors. The reason for this change is a recognition that a noise burst or other event might cause a sequence of errors, a condition for which the HEC is insufficient for error correction. The receiver remains in detection mode as long as errored cells are received. When a header is examined and found not to be in error, the receiver switches back to correction mode. The flowchart of Figure 11.6 shows the consequence of errors in the cell header.

The error-protection function provides both recovery from single-bit header errors, and a low probability of the delivery of cells with errored headers under bursty error conditions. The error characteristics of fiber-based transmission sys-

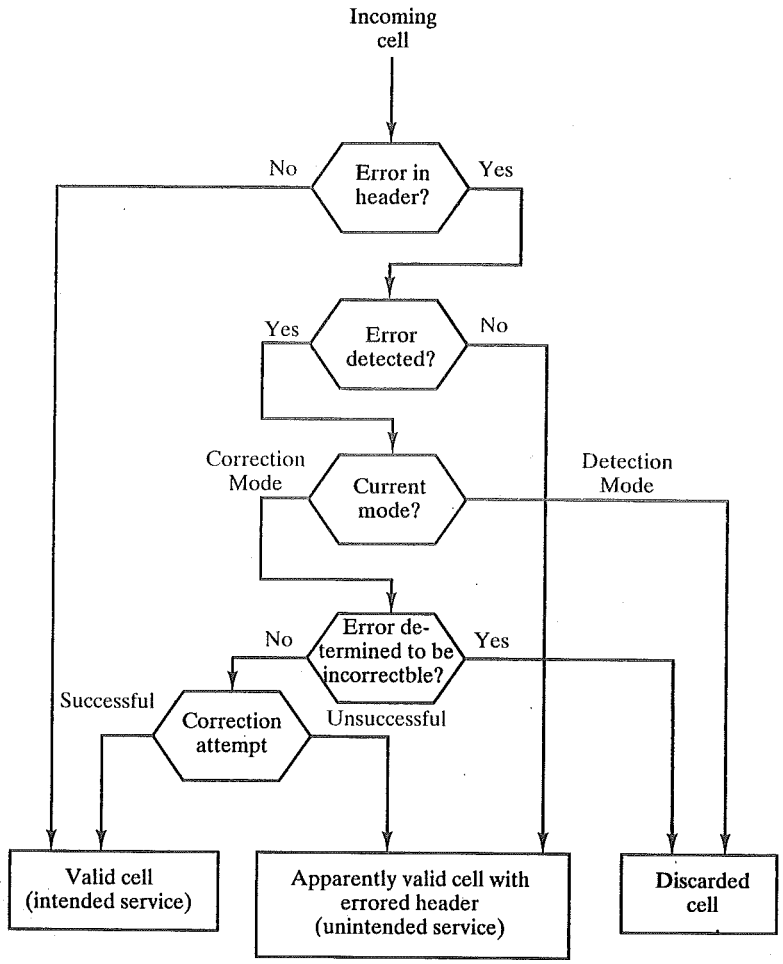


FIGURE 11.6 Effect of error in cell header.

tems appear to be a mix of single-bit errors and relatively large burst errors. For some transmission systems, the error-correction capability, which is more time-consuming, might not be invoked.

Figure 11.7, based on one in ITU-T I.432, indicates how random bit errors impact the probability of occurrence of discarded cells and valid cells with errored headers, when HEC is employed.

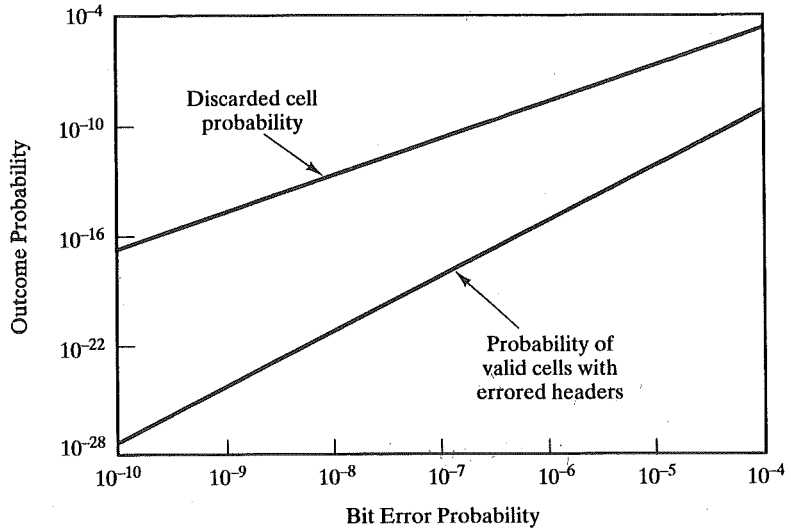


FIGURE 11.7 Impact of random bit errors on HEC performance.

11.4 TRANSMISSION OF ATM CELLS

The ITU-T Recommendations for broadband ISDN provide some detail on the data rate and synchronization techniques for ATM cell transmission across the user-network interface. The approach taken for broadband ISDN is also used in many other ATM networks.

The BISDN specifies that ATM cells are to be transmitted at a rate of 155.52 Mbps or 622.08 Mbps. As with ISDN, we need to specify the transmission structure that will be used to carry this payload. For 622.08 Mbps, the matter has been left for further study. For the 155.52-Mbps interface, two approaches are defined in I.413: a cell-based physical layer and an SDH-based physical layer. We examine each of these approaches in turn.

Cell-Based Physical Layer

For the cell-based physical layer, no framing is imposed. The interface structure consists of a continuous stream of 53-octet cells. Because there is no external frame

imposed on the cell-based approach, some form of synchronization is needed. Synchronization is achieved on the basis of the header error control (HEC) field in the cell header. The procedure is as follows (Figure 11.8):

1. In the HUNT state, a cell delineation algorithm is performed bit by bit to determine if the HEC coding law is observed (i.e., match between received HEC and calculated HEC). Once a match is achieved, it is assumed that one header has been found, and the method enters the PRESYNC state.
2. In the PRESYNC state, a cell structure is now assumed. The cell delineation algorithm is performed cell by cell until the encoding law has been confirmed consecutively δ times.
3. In the SYNC state, the HEC is used for error detection and correction (see Figure 11.5). Cell delineation is assumed to be lost if the HEC coding law is recognized as incorrect α times consecutively.

The values of α and δ are design parameters. Greater values of δ result in longer delays in establishing synchronization but in greater robustness against false delineation. Greater values of α result in longer delays in recognizing a misalignment but in greater robustness against false misalignment. Figures 11.9 and 11.10 show the impact of random bit errors on cell delineation performance for various values of α and δ. The first figure shows the average amount of time that the receiver will maintain synchronization in the face of errors, with α as a parameter. The second figure shows the average amount of time to acquire synchronization as a function of error rate, with δ as a parameter.

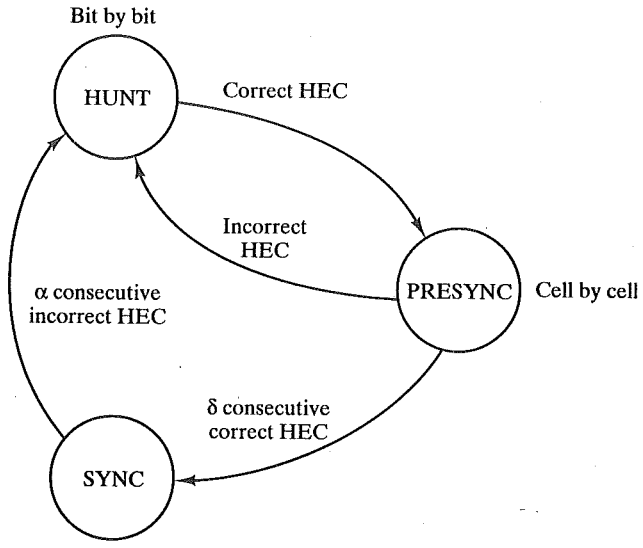


FIGURE 11.8 Cell delineation state diagram.

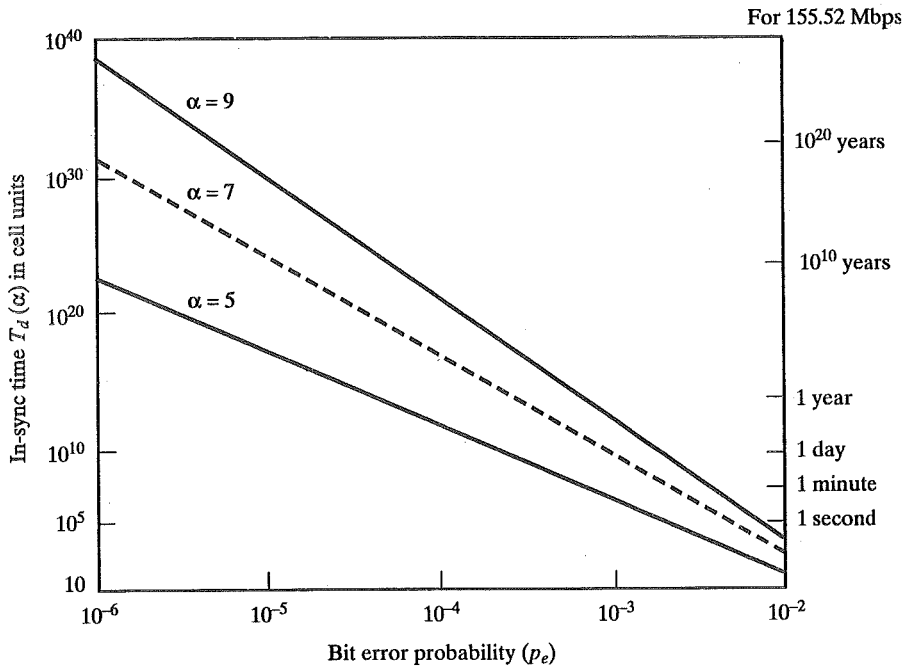


FIGURE 11.9 Impact of random bit errors on cell-delineation performance.

The advantage of using a cell-based transmission scheme is the simplified interface that results when both transmission- and transfer-mode functions are based on a common structure.

SDH-Based Physical Layer

Alternatively, ATM cells can be carried over a line using SDH (synchronous digital hierarchy) or SONET. For the cell-based physical layer, framing is imposed using the STM-1 (STS-3) frame. Figure 11.11 shows the payload portion of an STM-1 frame. This payload may be offset from the beginning of the frame, as indicated by the pointer in the section overhead of the frame. As can be seen, the payload consists of a 9-octet path overhead portion and the remainder, which contains ATM cells. Because the payload capacity (2,340 octets) is not an integer multiple of the cell length (53 octets), a cell may cross a payload boundary.

The H4 octet in the path overhead is set at the sending side to indicate the next occurrence of a cell boundary. That is, the value in the H4 field indicates the number of octets in the first cell boundary following the H4 octet. The permissible range of values is 0 to 52.

The advantages of the SDH-based approach include the following:

- It can be used to carry either ATM-based or STM-based (synchronous transfer mode) payloads, making it possible to initially deploy a high-capacity

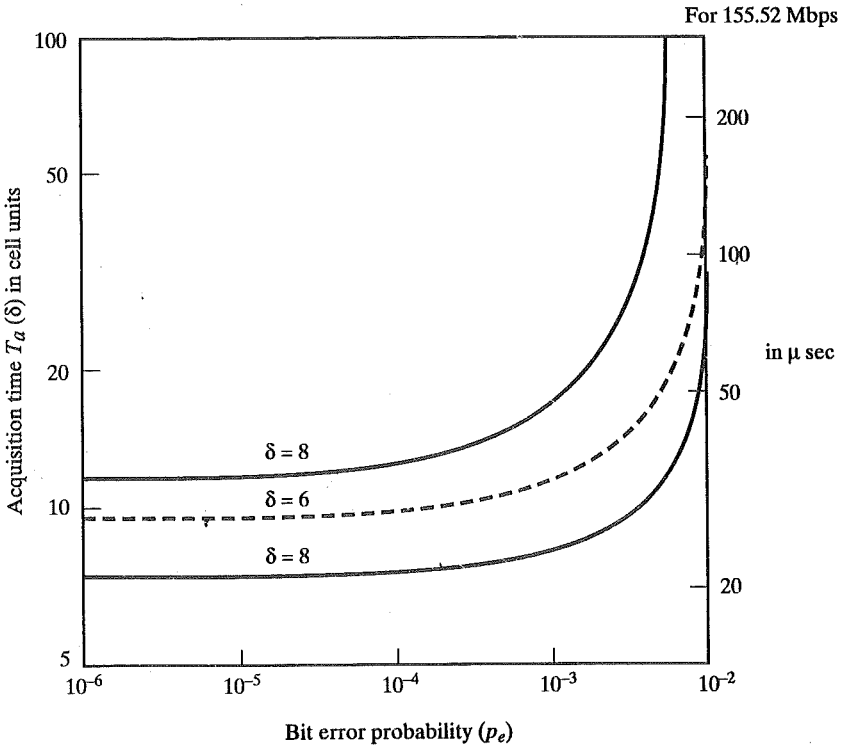


FIGURE 11.10 Acquisition time versus bit-error probability.

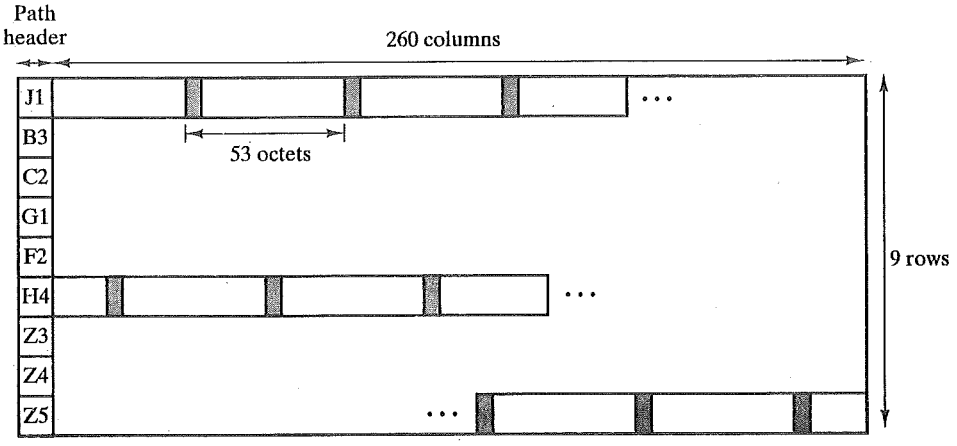


FIGURE 11.11 STM-1 payload for SDH-based ATM cell transmission.

fiber-based transmission infrastructure for a variety of circuit-switched and dedicated applications, and then readily migrate to the support of ATM.

- Some specific connections can be circuit-switched using an SDH channel. For example, a connection carrying constant-bit-rate video traffic can be mapped into its own exclusive payload envelope of the STM-1 signal, which can be circuit switched. This procedure may be more efficient than ATM switching.
- Using SDH synchronous multiplexing techniques, several ATM streams can be combined to build interfaces with higher bit rates than those supported by the ATM layer at a particular site. For example, four separate ATM streams, each with a bit rate of 155 Mbps (STM-1), can be combined to build a 622-Mbps (STM-4) interface. This arrangement may be more cost effective than one using a single 622-Mbps ATM stream.

11.5 ATM ADAPTATION LAYER

The use of ATM creates the need for an adaptation layer to support information transfer protocols not based on ATM. Two examples are PCM (pulse code modulation) voice and LAPF. PCM voice is an application that produces a stream of bits from a voice signal. To employ this application over ATM, it is necessary to assemble PCM bits into cells for transmission and to read them out on reception in such a way as to produce a smooth, constant flow of bits to the receiver. LAPF is the standard data link control protocol for frame relay. In a mixed environment, in which frame relay networks interconnect with ATM networks, a convenient way of integrating the two is to map LAPF frames into ATM cells; this will usually mean segmenting one LAPF frame into a number of cells on transmission, and then reassembling the frame from cells on reception. By allowing the use of LAPF over ATM, all of the existing frame relay applications and control signaling protocols can be used on an ATM network.

AAL Services

ITU-T I.362 lists the following general examples of services provided by AAL:

- Handling of transmission errors
- Segmentation and reassembly, to enable larger blocks of data to be carried in the information field of ATM cells
- Handling of lost and misinserted cell conditions
- Flow control and timing control

In order to minimize the number of different AAL protocols that must be specified to meet a variety of needs, ITU-T has defined four classes of service that cover a broad range of requirements (Figure 11.12). The classification is based on

	Class A	Class B	Class C	Class D
Timing relation between source and destination	Required		Not required	
Bit rate	Constant	Variable		
Connection mode	Connection-oriented			Connectionless
AAL Protocol	Type 1	Type 2	Type 3/4,Type 5	Type 3/4

FIGURE 11.12 Service classification for AAL.

whether a timing relationship must be maintained between source and destination, whether the application requires a constant bit rate, and whether the transfer is connection-oriented or connectionless. An example of a class A service is circuit emulation. In this case, a constant bit rate, which requires the maintenance of a timing relation, is used, and the transfer is connection-oriented. An example of a class B service is variable-bit-rate video, such as might be used in a videoconference. Here, the application is connection oriented and timing is important, but the bit rate varies depending on the amount of activity in the scene. Classes C and D correspond to data-transfer applications. In both cases, the bit rate may vary and no particular timing relationship is required; differences in data rate are handled, using buffers, by the end systems. The data transfer may be either connection-oriented (class C) or connectionless (class D).

AAL Protocols

To support these various classes of service, a set of protocols at the AAL level have been defined. The AAL layer is organized into two logical sublayers: the Convergence Sublayer (CS) and the Segmentation and Reassembly Sublayer (SAR). The convergence sublayer provides the functions needed to support specific applications using AAL. Each AAL user attaches to AAL at a service access point (SAP), which is simply the address of the application. This sublayer is, then, service dependent.

The segmentation and reassembly sublayer is responsible for packaging information received from CS into cells for transmission and unpacking the information at the other end. As we have seen, at the ATM layer, each cell consists of a 5-octet header and a 48-octet information field. Thus, SAR must pack any SAR headers and trailers, plus CS information, into 48-octet blocks.

Initially, ITU-T defined one protocol type for each class of service, named Type 1 through Type 4. Actually, each protocol type consists of two protocols, one at the CS sublayer and one at the SAR sublayer. More recently, types 3 and 4 were merged into a Type 3/4, and a new type, Type 5, was defined. Figure 11.12 shows which services are supported by which types. In all of these cases, a block of data from a higher layer is encapsulated into a protocol data unit (PDU) at the CS

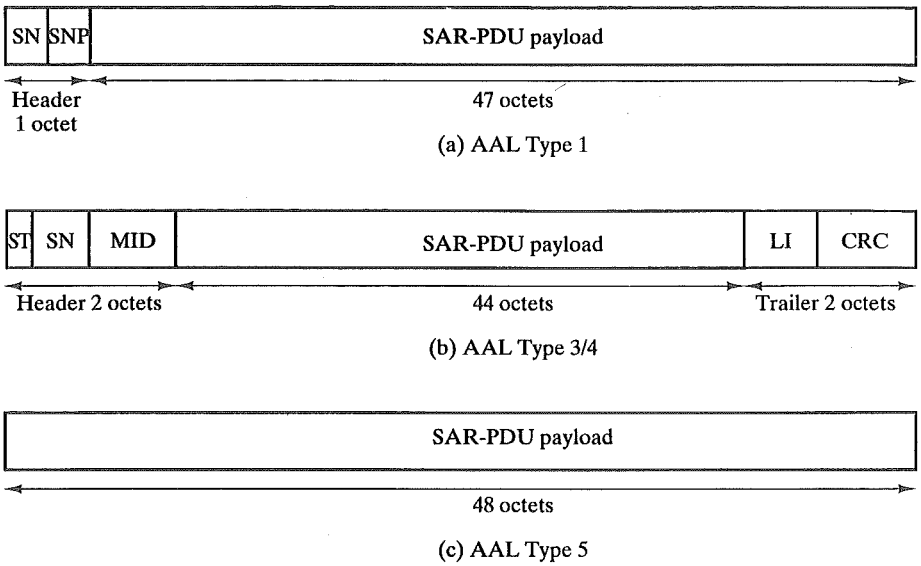


FIGURE 11.13 Segmentation and reassembly (SAR) protocol data units (PDUs).

sublayer. In fact, this sublayer is referred to as the common-part convergence sublayer (CPCS), leaving open the possibility that additional, specialized functions may be performed at the CS level. The CPCS PDU is then passed to the SAR sublayer, where it is broken up into payload blocks. Each payload block can fit into an SAR-PDU, which has a total length of 48 octets. Each 48-octet SAR-PDU fits into a single ATM cell.

Figure 11.13 shows the formats of the protocol data units (PDUs) at the SAR level except for Type 2, which has not yet been defined.

In the remainder of this section, we look at AAL Type 5, which is becoming increasingly popular, especially in ATM LAN applications. This protocol was introduced to provide a streamlined transport facility for higher-layer protocols that are connection-oriented. If it is assumed that the higher layer takes care of connection management, and that the ATM layer produces minimal errors, then most of the fields in the SAR and CPCS PDUs are not necessary. For example, with connection-oriented service, the MID field is not necessary. This field is used in AAL 3/4 to multiplex different streams of data using the same virtual ATM connection (VCI/VPI). In AAL 5, it is assumed that higher-layer software takes care of such multiplexing.

Type 5 was introduced to

- Reduce protocol-processing overhead
- Reduce transmission overhead
- Ensure adaptability to existing transport protocols

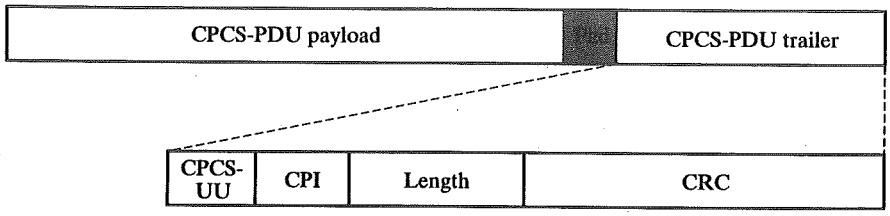
To understand the operation of Type 5, let us begin with the CPCS level. The CPCS-PDU (Figure 11.14) includes a trailer with the following fields:

- **CPCS User-to-User Indication** *(1 octet).* Used to transparently transfer user-to-user information.
- **Cyclic Redundancy Check** *(4 octets).* Used to detect bit errors in the CPCS-PDU.
- **Common Part Indicator** *(1 octet).* Indicates the interpretation of the remaining fields in the CPCS-PDU header. Currently, only one interpretation is defined.
- **Length** *(2 octets).* Length of the CPCS-PDU payload field.

The payload from the next higher layer is padded out so that the entire CPCS-PDU is a multiple of 48 octets.

The SAR-PDU consists simply of 48 octets of payload, carrying a portion of the CPCS-PDU. The lack of protocol overhead has several implications:

1. Because there is no sequence number, the receiver must assume that all SAR-PDUs arrive in the proper order for reassembly. The CRC field in the CPCS-PDU is intended to verify such an order.
2. The lack of MID field means that it is not possible to interleave cells from different CPCS-PDUs. Therefore, each successive SAR-PDU carries a portion



LEGEND

CPCS-UU = CPCS user-to-user indication (1 octet)
CPI = Common-part indicator (1 octet)
Length = Length of CPCS-PDU payload (2 octets)
CRC = Cyclic redundancy check (4 octets)

FIGURE 11.14 AAL type 5 CPCS PDU.

of the current CPCS-PDU, or the first block of the next CPCS-PDU. To distinguish between these two cases, the ATM user-to-user indication (AAU) bit in the payload-type field of the ATM cell header is used (Figure 11.4). A CPCS-PDU consists of zero or more consecutive SAR-PDUs with AAU set to 0, followed immediately by an SAR-PDU with AAU set to 1.

3. The lack of an LI field means that there is no way for the SAR entity to distinguish between CPCS-PDU octets and filler in the last SAR-PDU. Therefore, there is no way for the SAR entity to find the CPCS-PDU trailer in the last SAR-PDU. To avoid this situation, it is required that the CPCS-PDU payload be padded out so that the last bit of the CPCS-trailer occurs as the last bit of the final SAR-PDU.

Figure 11.15 shows an example of AAL 5 transmission. The CPCS-PDU, including padding and trailer, is divided into 48-octet blocks. Each block is transmitted in a single ATM cell.

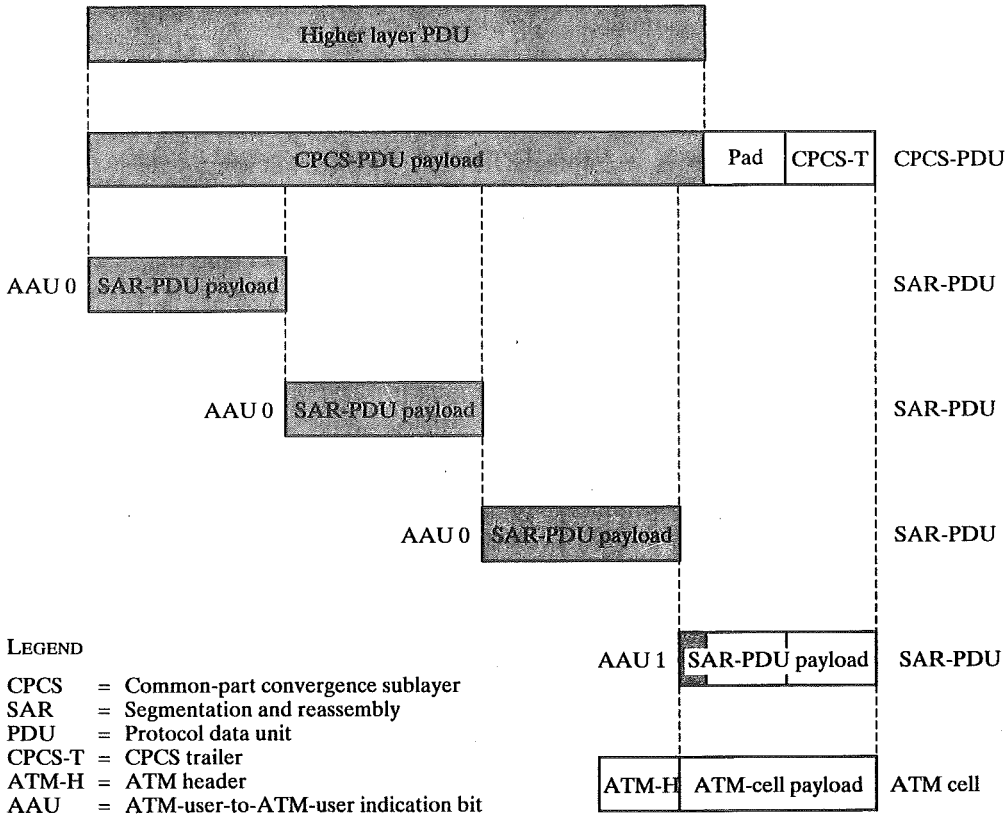


FIGURE 11.15 Example of AAL 5 transmission.

11.6 TRAFFIC AND CONGESTION CONTROL

As is the case with frame relay networks, traffic and congestion control techniques are vital to the successful operation of ATM-based networks. Without such techniques, traffic from user nodes can exceed the capacity of the network, causing memory buffers of ATM switches to overflow, leading to data losses.

ATM networks present difficulties in effectively controlling congestion not found in other types of networks, including frame relay networks. The complexity of the problem is compounded by the limited number of overhead bits available for exerting control over the flow of user cells. This area is currently the subject of intense research, and no consensus has emerged for a full-blown traffic- and congestion-control strategy. Accordingly, ITU-T has defined a restricted initial set of traffic- and congestion-control capabilities aiming at simple mechanisms and realistic network efficiency; these are specified in I.371.

We begin with an overview of the congestion problem and the framework adopted by ITU-T. We see that the focus of the mechanisms so far adopted is on control schemes for delay-sensitive traffic, such as voice and video. These schemes are not suited for handling bursty traffic, which is the subject of ongoing research and standardization efforts. The discussion then turns to traffic control, which refers to the set of actions taken by the network to avoid congestion. Finally, we examine congestion control, which refers to the set of actions taken by the network to minimize the intensity, spread, and duration of congestion once congestion has already occurred.

Requirements for ATM Traffic and Congestion Control

Both the types of traffic patterns imposed on ATM network and the transmission characteristics of those network differ markedly from those of other switching networks. Most packet-switched and frame relay networks carry non-real-time data traffic. Typically, the traffic on individual virtual circuits or frame relay connections is bursty in nature, and the receiving system expects to receive incoming traffic on each connection in such a fashion. As a result,

1. The network does not need to replicate the exact timing pattern of incoming traffic at the exit node.
2. Therefore, simple statistical multiplexing can be used to accommodate multiple logical connections over the physical interface between user and network. The average data rate required by each connection is less than the burst rate for that connection, and the user-network interface (UNI) need only be designed for a capacity somewhat greater than the sum of the average data rates for all connections.

A number of tools are available for control of congestion in packet-switched and frame relay networks, as we have seen in the preceding two chapters. These types of congestion-control schemes are inadequate for ATM networks. [GERS91] cites the following reasons:

1. The majority of traffic is not amenable to flow control. For example, voice and video traffic sources cannot stop generating cells even when the network is congested.
2. Feedback is slow due to the drastically reduced cell transmission time compared to propagation delays across the network.
3. ATM networks typically support a wide range of applications requiring capacity ranging from a few kbps to several hundred Mbps. Relatively simple-minded congestion control schemes generally end up penalizing one end or the other of that spectrum.
4. Applications on ATM networks may generate very different traffic patterns (e.g., constant bit-rate versus variable bit-rate sources). Again, it is difficult for conventional congestion control techniques to handle fairly such variety.
5. Different applications on ATM networks require different network services (e.g., delay-sensitive service for voice and video, and loss-sensitive service for data).
6. The very high speeds in switching and transmission make ATM networks more volatile in terms of congestion and traffic control. A scheme that relies heavily on reacting to changing conditions will produce extreme and wasteful fluctuations in routing policy and flow control.

A key issue that relates to the above points is cell delay variation, a topic to which we now turn.

Cell-Delay Variation

For an ATM network, voice and video signals can be digitized and transmitted as a stream of cells. A key requirement, especially for voice, is that the delay across the network be short; generally, this will be the case for ATM networks. As we have discussed, ATM is designed to minimize the processing and transmission overhead internal to the network so that very fast cell switching and routing are possible.

There is another important requirement that, to some extent, conflicts with the preceding requirement, namely that the rate of delivery of cells to the destination user must be constant. Now, it is inevitable that there will be some variability in the rate of delivery of cells, due both to effects within the network and at the source UNI; we summarize these effects presently. First, let us consider how the destination user might cope with variations in the delay of cells as they transit from source user to destination user.

A general procedure for achieving a constant bit rate (CBR) is illustrated in Figure 11.16. Let $D(i)$ represent the end-to-end delay experienced by the ith cell. The destination system does not know the exact amount of this delay; there is no timestamp information associated with each cell, and, even if there were, it is impossible to keep source and destination clocks perfectly synchronized. When the first cell on a connection arrives at time $t(0)$, the target user delays the cell an additional amount $V(0)$ prior to delivery to the application. $V(0)$ is an estimate of the

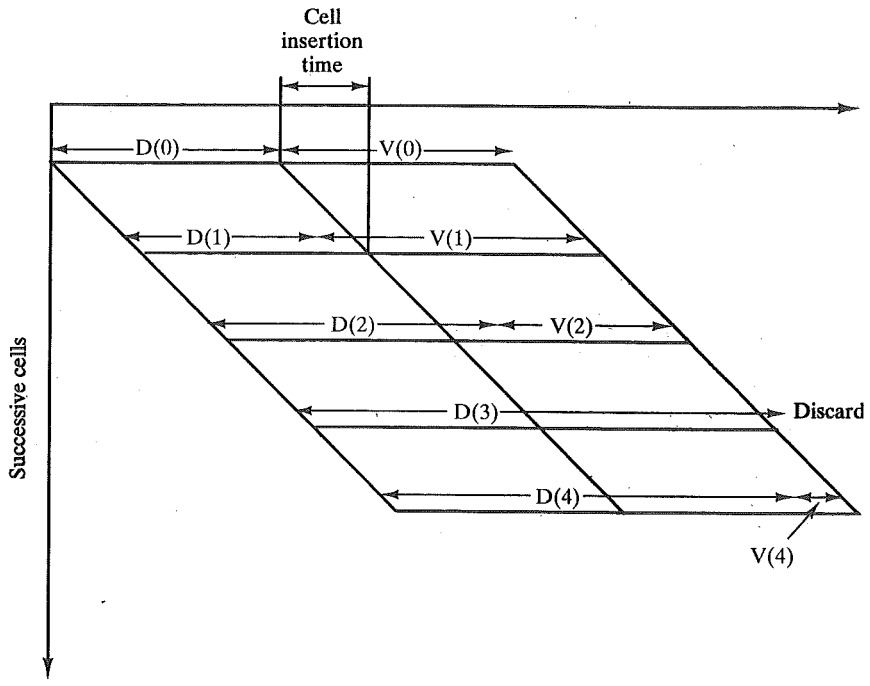


FIGURE 11.16 Time reassembly of CBR cells.

amount of cell delay variation that this application can tolerate and that is likely to be produced by the network.

Subsequent cells are delayed so that they are delivered to the user at a constant rate of R cells per second. The time between delivery of cells to the target application is therefore $\delta = 1/R$. To achieve a constant rate, the next cell is delayed a variable amount $V(1)$ to satisfy the following:

$$t(1) + V(1) = t(0) + V(0) + \delta$$

So,

$$V(1) = V(0) - [t(1) - (t(0) + \delta)]$$

In general,

$$V(i) = V(0) - [t(i) - (t(0) + i \times \delta)]$$

which can also be expressed as

$$V(i) = V(i-1) - [t(i) - (t(i-1) + \delta)]$$

If the computed value of $V(i)$ is negative, then that cell is discarded. The result is that data is delivered to the higher layer at a constant bit rate, with occasional gaps due to dropped cells.

The amount of the initial delay $V(0)$, which is also the average delay applied to all incoming cells, is a function of the anticipated cell-delay variation. To minimize this delay, a subscriber will therefore request a minimal cell-delay variation from the network provider. This request leads to a trade-off; cell-delay variation can be reduced by increasing the data rate at the UNI, relative to the load, and by increasing resources within the network.

Network Contribution to Cell-Delay Variation

One component of cell-delay variation is due to events within the network. For packet-switching networks, packet delay variation can be considerable, due to queuing effects at each of the intermediate switching nodes; to a lesser extent, this is also true of frame delay variation in frame relay networks. However, in the case of ATM networks, cell-delay variations due to network effects are likely to be minimal; the principal reasons for this are the following:

1. The ATM protocol is designed to minimize processing overhead at intermediate switching nodes. The cells are fixed-size with fixed-header formats, and there is no flow control or error control processing required.
2. To accommodate the high speeds of ATM networks, ATM switches have had to be designed to provide extremely high throughput. Thus, the processing time for an individual cell at a node is negligible.

The only factor that could lead to noticeable cell-delay variation within the network is congestion. If the network begins to become congested, either cells must be discarded or there will be a buildup of queuing delays at affected switches. Thus, it is important that the total load accepted by the network at any time not be such as to cause congestion.

Cell-Delay Variation at the UNI

Even if an application generates data for transmission at a constant bit rate, cell-delay variation can occur at the source due to the processing that takes place at the three layers of the ATM model.

Figure 11.17 illustrates the potential causes of cell-delay variation. In this example, ATM connections A and B support user data rates of X and Y Mbps, respectively. At the AAL level, data is segmented into 48-octet blocks. Note that on a time diagram, the blocks appear to be of different sizes for the two connections; specifically, the time required to generate a 48-octet block of data in microseconds is

$$\text{Connection A: } \frac{48 \times 8}{X}$$

$$\text{Connection B: } \frac{48 \times 8}{Y}$$

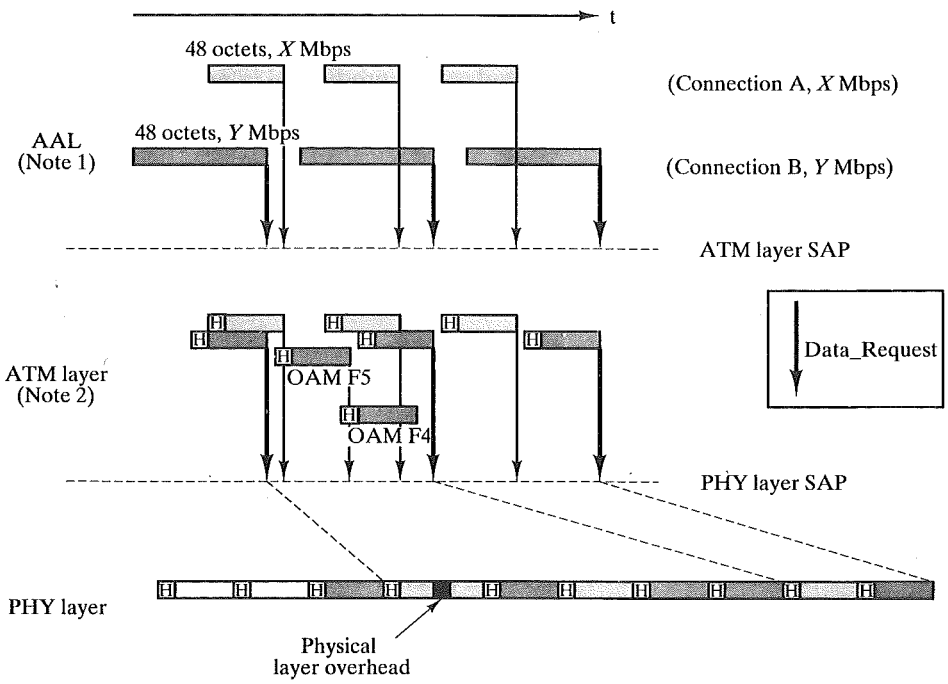


FIGURE 11.17 Origins of cell delay variation (I.371).

The ATM layer encapsulates each segment into a 53-octet cell. These cells must be interleaved and delivered to the physical layer to be transmitted at the data rate of the physical link. Delay is introduced into this interleaving process: If two cells from different connections arrive at the ATM layer at overlapping times, one of the cells must be delayed by the amount of the overlap. In addition, the ATM layer is generating OAM cells that must also be interleaved with user cells.

At the physical layer, there is additional opportunity for the introduction of further cell delays. For example, if cells are transmitted in SDH frames, overhead

TABLE 11.3 Traffic control and congestion control functions.

Response time	Traffic control functions	Congestion control functions
Long Term	• Network resource management	
Connection Duration	• Connection admission control	
Round-trip Propagation Time	• Fast resource management	• Explicit notification
Cell Insertion Time	• Usage parameter control • Priority control	• Selective cell discarding

bits for those frames will be inserted into the physical link, thereby delaying bits from the ATM layer.

None of the delays just listed can be predicated in any detail, and none follow any repetitive pattern. Accordingly, there is a random element to the time interval between reception of data at the ATM layer from the AAL and the transmission of that data in a cell across the UNI.

Traffic and Congestion Control Framework

I.371 lists the following objectives of ATM layer traffic and congestion control:

- ATM layer traffic and congestion control should support a set of ATM layer Quality of Service (QOS) classes sufficient for all foreseeable network services; the specification of these QOS classes should be consistent with network performance parameters currently under study.
- ATM layer traffic and congestion control should not rely on AAL protocols that are network-service specific, nor on higher-layer protocols that are application specific. Protocol layers above the ATM layer may make use of information provided by the ATM layer to improve the utility those protocols can derive from the network.
- The design of an optimum set of ATM layer traffic controls and congestion controls should minimize network and end-system complexity while maximizing network utilization.

In order to meet these objectives, ITU-T has defined a collection of traffic and congestion control functions that operate across a spectrum of timing intervals. Table 11.3 lists these functions with respect to the response times within which they operate. Four levels of timing are considered:

- **Cell insertion time.** Functions at this level react immediately to cells as they are transmitted.
- **Round-trip propagation time.** At this level, the network responds within the lifetime of a cell in the network, and may provide feedback indications to the source.
- **Connection duration.** At this level, the network determines whether a new connection at a given QOS can be accommodated and what performance levels will be agreed to.
- **Long term.** These are controls that affect more than one ATM connection and that are established for long-term use.

The essence of the traffic-control strategy is based on (1) determining whether a given new ATM connection can be accommodated and (2) agreeing with the subscriber on the performance parameters that will be supported. In effect, the subscriber and the network enter into a traffic contract: The network agrees to support traffic at a certain level on this connection, and the subscriber agrees not to exceed performance limits. Traffic control functions are concerned with establishing these traffic parameters and enforcing them. Thus, they are concerned with congestion

avoidance. If traffic control fails in certain instances, then congestion may occur. At this point, congestion-control functions are invoked to respond to and recover from the congestion.

Traffic Control

A variety of traffic control functions have been defined to maintain the QOS of ATM connections. These include

- Network resource management
- Connection admission control
- Usage parameter control
- Priority control
- Fast resource management

We examine each of these in turn.

Network Resource Management

The essential concept behind network resource management is to allocate network resources in such a way as to separate traffic flows according to service characteristics. So far, the only specific traffic control function based on network resource management defined by ITU-T Forum deals with the use of virtual paths.

As discussed earlier, a virtual path connection (VPC) provides a convenient means of grouping similar virtual channel connections (VCCs). The network provides aggregate capacity and performance characteristics on the virtual path, and these are shared by the virtual connections. There are three cases to consider:

- **User-to-user application.** The VPC extends between a pair of UNIs. In this case, the network has no knowledge of the QOS of the individual VCCs within a VPC. It is the user's responsibility to assure that the aggregate demand from the VCCs can be accommodated by the VPC.
- **User-to-network application.** The VPC extends between a UNI and a network node. In this case, the network is aware of the QOS of the VCCs within the VPC and has to accommodate them.
- **Network-to-network application.** The VPC extends between two network nodes. Again, in this case, the network is aware of the QOS of the VCCs within the VPC and has to accommodate them.

The QOS parameters that are of primary concern for network resource management are cell loss ratio, cell transfer delay, and cell delay variation, all of which are affected by the number of resources devoted to the VPC by the network. If a VCC extends through multiple VPCs, then the performance on that VCC depends on the performances of the consecutive VPCs, and on how the connection is handled at any node that performs VCC-related functions. Such a node may be a switch, concentrator, or other network equipment. The performance of each VPC depends on the capacity of that VPC and the traffic characteristics of the VCCs contained within the VPC. The performance of each VCC-related function depends on

the switching/processing speed at the node and on the relative priority with which various cells are handled.

Figure 11.18 gives an example. VCCs 1 and 2 experience a performance that depends on VPCs *b* and *c* and on how these VCCs are handled by the intermediate nodes; this may differ from the performance experienced by VCCs 3, 4, and 5.

There are a number of alternatives for the way in which VCCs are grouped and the type of performance they experience. If all of the VCCs within a VPC are handled similarly, then they should experience similar expected network performance, in terms of cell-loss ratio, cell-transfer delay, and cell-delay variation. Alternatively, when different VCCs within the same VPC require different QOS, the VPC performance objective agreed upon by network and subscriber should be suitably set for the most demanding VCC requirement.

In either case, with multiple VCCs within the same VPC, the network has two general options for allocating capacity to the VPC:

Aggregate peak demand. The network may set the capacity (data rate) of the VPC equal to the total of the peak data rates of all of the VCCs within the VPC. The advantage of this approach is that each VCC can be given a QOS that accommodates its peak demand. The disadvantage is that most of the time, the VPC capacity will not be fully utilized, and, therefore, the network will have underutilized resources.

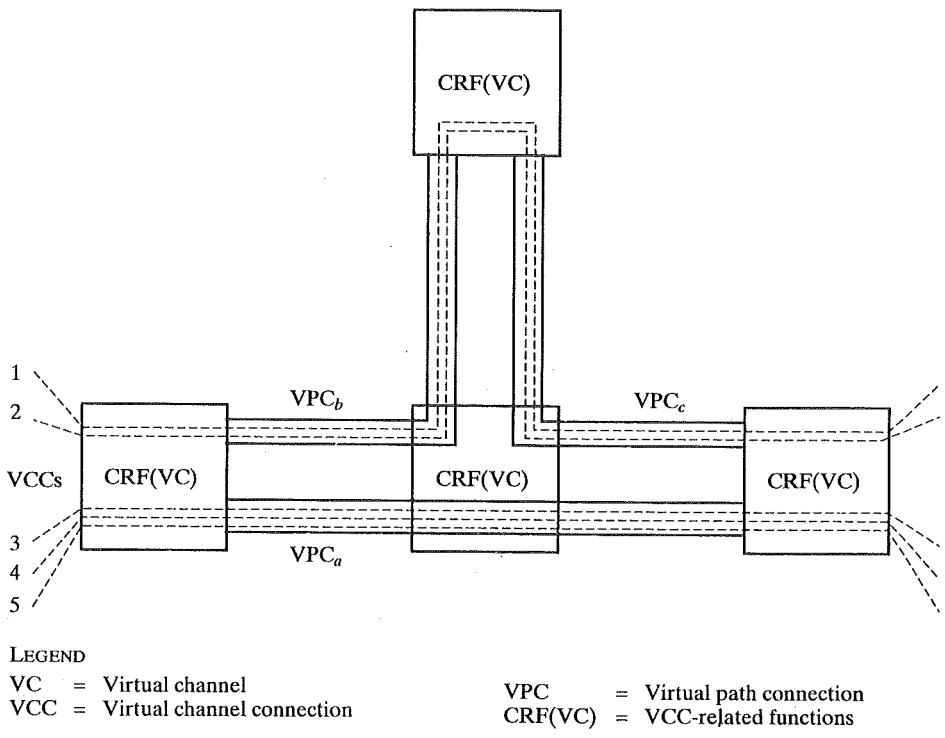


FIGURE 11.18 Configuration of VCCs and VPCs (I.371).

2. *Statistical multiplexing.* If the network sets the capacity of the VPC to be greater than or equal to the average data rates of all the VCCs but less than the aggregate peak demand, then a statistical multiplexing service is supplied. With statistical multiplexing, VCCs experience greater cell-delay variation and greater cell-transfer delay. Depending on the size of buffers used to queue cells for transmission, VCCs may also experience greater cell-loss ratio. This approach has the advantage of more efficient utilization of capacity, and is attractive if the VCCs can tolerate the lower QOS.

When statistical multiplexing is used, it is preferable to group VCCs into VPCs on the basis of similar traffic characteristics and similar QOS requirements. If dissimilar VCCs share the same VPC and statistical multiplexing is used, it is difficult to provide fair access to both high-demand and low-demand traffic streams.

Connection Admission Control

Connection admission control is the first line of defense for the network in protecting itself from excessive loads. In essence, when a user requests a new VPC or VCC, the user must specify (implicitly or explicitly) the traffic characteristics in both directions for that connection. The user selects traffic characteristics by selecting a QOS from among the QOS classes that the network provides. The network accepts the connection only if it can commit the resources necessary to support that traffic level while at the same time maintaining the agreed-upon QOS of existing connections. By accepting the connection, the network forms a *traffic contract* with the user. Once the connection is accepted, the network continues to provide the agreed-upon QOS as long as the user complies with the traffic contract.

For the current specification, the traffic contract consists of the four parameters defined in Table 11.4: peak cell rate (PCR), cell-delay variation (CDV), sustainable

TABLE 11.4 Traffic parameters used in defining VCC/VPC quality of service.

Parameter	Description	Traffic type
Peak Cell Rate (PCR)	An upper bound on the traffic that can be submitted on an ATM connection.	CBR, VBR
Cell Delay Variation (CDV)	An upper bound on the variability in the pattern of cell arrivals observed at a single measurement point with reference to the peak cell rate.	CBR, VBR
Sustainable Cell Rate (SCR)	An upper bound on the average rate of an ATM connection, calculated over the duration of the connection.	VBR
Burst Tolerance	An upper bound on the variability in the pattern of cell arrivals observed at a single measurement point with reference to the sustainable cell rate.	VBR

CBR = constant bit rate
VBR = variable bit rate

cell rate (SCR), and burst tolerance. Only the first two parameters are relevant for a constant bit rate (CBR) source; all four parameters may be used for variable bit rate (VBR) sources.

As the name suggests, the peak cell rate is the maximum rate at which cells are generated by the source on this connection. However, we need to take into account the cell-delay variation. Although a source may be generating cells at a constant peak rate, cell-delay variations introduced by various factors (see Figure 11.17) will affect the timing, causing cells to clump up and gaps to occur. Thus, a source may temporarily exceed the peak cell rate due to clumping. For the network to properly allocate resources to this connection, it must know not only the peak cell rate but also the CDV.

The exact relationship between peak cell rate and CDV depends on the operational definitions of these two terms. The standards provide these definitions in terms of a cell rate algorithm. Because this algorithm can be used for usage parameter control, we defer a discussion until the next subsection.

The PCR and CDV must be specified for every connection. As an option for variable-bit rate sources, the user may also specify a sustainable cell rate and burst tolerance. These parameters are analogous to PCR and CDV, respectively, but apply to an average rate of cell generation rather than to a peak rate. The user can describe the future flow of cells in greater detail by using the SCR and burst tolerance as well as the PCR and CDV. With this additional information, the network may be able to more efficiently utilize the network resources. For example, if a number of VCCs are statistically multiplexed over a VPC, knowledge of both average and peak cell rates enables the network to allocate buffers of sufficient size to handle the traffic efficiently without cell loss.

For a given connection (VPC or VCC), the four traffic parameters may be specified in several ways, as illustrated in Table 11.5. Parameter values may be implicitly defined by default rules set by the network operator. In this case, all connections are assigned the same values or all connections of a given class are assigned

TABLE 11.5 Procedures used to set values of traffic contract parameters.

	Explicitly specified parameters		Implicitly specified parameters
	Parameter values set at connection-setup time	Parameter values specified at subscription time	Parameter values set using default rules
	Requested by user/NMS	assigned by network operator	
SVC	signaling	by subscription	network-operator default rules
PVC	NMS	by subscription	network-operator default rules

SVC = switched virtual connection
PVC = permanent virtual connection
NMS = network management system

the same values for that class. The network operator may also associate parameter values with a given subscriber and assign these at the time of subscription. Finally, parameter values tailored to a particular connection may be assigned at connection time. In the case of a permanent virtual connection, these values are assigned by the network when the connection is set up. For a switched virtual connection, the parameters are negotiated between the user and the network via a signaling protocol.

Another aspect of quality of service that may be requested or assigned for a connection is cell-loss priority. A user may request two levels of cell-loss priority for an ATM connection; the priority of an individual cell is indicated by the user through the CLP bit in the cell header (see Figure 11.4). When two priority levels are used, the traffic parameters for both cell flows must be specified; typically, this is done by specifying a set of traffic parameters for high-priority traffic (CLP = 0) and a set of traffic parameters for all traffic (CLP = 0 or 1). Based on this breakdown, the network may be able to allocate resources more efficiently.

Usage Parameter Control

Once a connection has been accepted by the Connection Admission Control function, the Usage Parameter Control (UPC) function of the network monitors the connection to determine whether the traffic conforms to the traffic contract. The main purpose of Usage Parameter Control is to protect network resources from an overload on one connection that would adversely affect the QOS on other connections by detecting violations of assigned parameters and taking appropriate actions.

Usage parameter control can be done at both the virtual path and virtual channel levels. Of these, the more important is VPC-level control, as network resources are, in general, initially allocated on the basis of virtual paths, with the virtual path capacity shared among the member virtual channels.

There are two separate functions encompassed by usage parameter control:

- Control of peak cell rate and the associated cell-delay variation (CDV)
- Control of sustainable cell rate and the associated burst tolerance

Let us first consider the peak cell rate and the associated cell-delay variation. In simple terms, a traffic flow is compliant if the peak rate of cell transmission does not exceed the agreed-upon peak cell rate, subject to the possibility of cell-delay variation within the agreed-upon bound. I.371 defines an algorithm, the peak cell-rate algorithm, that monitors compliance. The algorithm operates on the basis of two parameters: a peak cell-rate R and a CDV tolerance limit of τ. Then, $T = 1/R$ is the interarrival time between cells if there were no CDV. With CDV, T is the average interarrival time at the peak rate. The algorithm uses a form of leaky-bucket mechanism to monitor the rate at which cells arrive in order to assure that the interarrival time is not too short to cause the flow to exceed the peak cell rate by an amount greater than the tolerance limit.

The same algorithm, with different parameters can be used to monitor the sustainable cell rate and the associated burst tolerance. In this case, the parameters are the sustainable cell-rate R_s and a burst tolerance τ_s.

The cell-rate algorithm is rather complex; details can be found in [STAL95a]. The cell-rate algorithm simply defines a way to monitor compliance with the traffic

contract. To perform usage parameter control, the network must act on the results of the algorithm. The simplest strategy passes along compliant cells and discards noncompliant cells at the point of the UPC function.

At the network's option, cell tagging may also be used for noncompliant cells. In this case, a noncompliant cell may be tagged with CLP = 1 (low priority) and passed. Such cells are then subject to discard at a later point in the network.

If the user has negotiated two levels of cell-loss priority for a network, then the situation is more complex. Recall that the user may negotiate a traffic contract for high-priority traffic (CLP = 0) and a separate contract for aggregate traffic (CLP 0 or 1). The following rules apply:

1. A cell with CLP = 0 that conforms to the traffic contract for CLP = 0 passes.
2. A cell with CLP = 0 that is noncompliant for (CLP = 0) traffic but compliant for (CLP 0 or 1) traffic is tagged and passed.
3. A cell with CLP = 0 that is noncompliant for (CLP = 0) traffic and noncompliant for (CLP 0 or 1) traffic is discarded.
4. A cell with CLP = 1 that is compliant for (CLP = 1) traffic is passed.
5. A cell with CLP = 1 that is noncompliant for (CLP 0 or 1) traffic is discarded.

Priority Control

Priority control comes into play when the network, at some point beyond the UPC function, discards (CLP = 1) cells. The objective is to discard lower priority cells in order to protect the performance for higher-priority cells. Note that the network has no way to discriminate between cells that were labeled as lower-priority by the source and cells that were tagged by the UPC function.

Fast Resource Management

Fast resource management functions operate on the time scale of the round-trip propagation delay of the ATM connection. The current version of I.371 lists fast-resource management as a potential tool for traffic control that is for further study. One example of such a function that is given in the Recommendation is the ability of the network to respond to a request by a user to send a burst. That is, the user would like to temporarily exceed the current traffic contract to send a relatively large amount of data. If the network determines that the resources exist along the route for this VCC or VPC for such a burst, then the network reserves those resources and grants permission. Following the burst, the normal traffic control is enforced.

Congestion Control

ATM congestion control refers to the set of actions taken by the network to minimize the intensity, spread, and duration of congestion. These actions are triggered by congestion in one or more network elements. The following two functions have been defined:

- Selective cell discarding
- Explicit forward congestion indication

Selective Cell Discarding

Selective cell discarding is similar to priority control. In the priority control function (CLP = 1), cells are discarded to avoid congestion. However, only "excess" cells are discarded. That is, cells are limited so that the performance objectives for the (CLP = 0) and (CLP = 1) flows are still met. Once congestion actually occurs, the network is no longer bound to meet all performance objectives. To recover from a congested condition, the network is free to discard any (CLP = 1) cell and may even discard (CLP = 0) cells on ATM connections that are not complying with their traffic contract.

Explicit Forward Congestion Indication

Explicit forward congestion notification for ATM network works in essentially the same manner as for frame relay networks. Any ATM network node that is experiencing congestion may set an explicit forward congestion indication in the payload type field of the cell header of cells on connections passing through the node (Figure 11.4). The indication notifies the user that congestion avoidance procedures should be initiated for traffic in the same direction as the received cell. It indicates that this cell on this ATM connection has encountered congested resources. The user may then invoke actions in higher-layer protocols to adaptively lower the cell rate of the connection.

The network issues the indication by setting the first two bits of the payload type field in the cell header to 01 (Table 11.2). Once this value is set by any node, it may not be altered by other network nodes along the path to the destination user.

Note that the generic flow control (GFC) field is not involved. The GFC field has only local significance and cannot be communicated across the network.

11.7 RECOMMENDED READING

[GORA95], [MCDY95], [HAND94], and [PRYC93] provide in-depth coverage of ATM. An interesting overview of ATM is [BOUD92]. The virtual path/virtual channel approach of ATM is examined in [SATO90], [SATO91], and [BURG91].

[ARMI93] and [SUZU94] discuss AAL and compare Types 3/4 and 5.

[ONVU94] is devoted to issues related to the performance of ATM networks, including traffic and congestion control. The following special issues are devoted to the topics of this chapter: April 1991 issue of IEEE Journal on Selected Areas in Communications; October 1991 issue of IEEE Communications Magazine; and September 1992 issue of IEEE Network.

ARMI93 Armitage, G. and Adams, K. "Packet Reassembly During Cell Loss." *IEEE Network*, September 1993.

BOUD92 Boudec, J. "The Asynchronous Transfer Mode: A Tutorial." *Computer Networks and ISDN Systems*, May 1992.

BURG91 Burg, J. and Dorman, D. "Broadband ISDN Resource Management: The Role of Virtual Paths." *IEEE Communications Magazine*, September 1991.

GORA95 Goralski, W. *Introduction to ATM Networking*. New York: McGraw-Hill, 1995.

HAND94 Handel, R., Huber, N., and Schroder, S. ATM Networks: *Concepts, Protocols, Applications*. Reading, MA: Addison-Wesley, 1994.

MCDY95 McDysan, D. and Spohn, D. *ATM: Theory and Application*. New York: McGraw-Hill, 1995.

ONVU94 Onvural, R. *Asynchronous Transfer Mode Networks: Performance Issues*. Boston: Artech House, 1994.

PRYC93 Prycker, M. *Asynchronous Transfer Mode: Solutions for Broadband ISDN*. New York: Ellis Horwood, 1993.

SATO90 Sato, K., Ohta, S., and Tokizawa, I. "Broad-band ATM Network Architecture Based on Virtual Paths." *IEEE Transactions on Communications*, August 1990.

SATO91 Sato, K., Ueda, H., and Yoshikai, M. "The Role of Virtual Path Crossconnection." *IEEE LTS*, August 1991.

SUZU94 Suzuki, T. "ATM Adaptation Layer Protocol." *IEEE Communications Magazine*, April 1994.

Recommended Web Sites

- http://www.atmforum.com: The web site of the ATM forum, which is leading the effort to expand the funtionality of ATM networks.
- http://www.atm25.com/ATM_Reference.html: Links to dozens of ATM reference sites on the Internet.

11.8 PROBLEMS

11.1 One method of transmitting ATM cells is as a continuous stream of cells, with no framing imposed; therefore, the transmission is simply a stream of bits, with all bits being part of cells. Because there is no external frame, some other form of synchronization is needed, and can be achieved using the HEC function. The requirement is to assure that the receiver knows the beginning and ending cell boundaries and does not drift with respect to the sender. Draw a state diagram for the use of the HEC to achieve cell synchronization, and then explain its functionality.

11.2 Although ATM does not include any end-to-end error detection and control functions on the user data, it is provided with an HEC field to detect and correct header errors. Let us consider the value of this feature. Suppose that the bit error rate of the transmission system is B. If errors are uniformly distributed, then the probability of an error in the header is

$$\frac{h}{h+i} \times B$$

and the probability of an error in the data field is

$$\frac{i}{h+i} \times B$$

where h is the number of bits in the header and i is the number of bits in the data field.

a. Suppose that errors in the header are not detected and not corrected. In this case, a header error may result in a misrouting of the cell to the wrong destination; therefore, i bits will arrive at an incorrect destination, and i bits will not arrive at the correct destination. What is the overall bit error rate $B1$? Find an expression for the multiplication effect on the bit error rate $M1 = B1/B$.

b. Now suppose that header errors are detected but not corrected. In that case, i bits will not arrive at the correct destination. What is the overall bit error rate $B2$? Find an expression for the multiplication effect on the bit error rate: $M2 = B2/B$.

c. Now suppose that header errors are detected and corrected. What is the overall bit error rate $B3$? Find an expression for the multiplication effect on the bit rate error $M3 = B3/B$.

d. Plot $M1$, $M2$, and $M3$ as a function of header length, for $i = 48 \times 8 = 384$ bits. Comment on the results.

11.3 One key design decision for ATM was whether to use fixed or variable length cells. Let us consider this decision from the point of view of efficiency. We can define transmission efficiency as

$$N = \frac{\text{Number of information octets}}{\text{Number of information octets} + \text{Number of overhead octets}}$$

a. Consider the use of fixed-length packets. In this case, the overhead consists of the header octets. Define the following:

L = Data-field size of the cell in octets
H = Header size of the cell in octets
X = Number of information octets to be transmitted as a single message

Derive an expression for N. Hint, the expression will need to use the operator $\lceil \cdot \rceil$, where

$\lceil Y \rceil$ = the smallest integer greater than or equal to Y.

b. If cells have variable length, then overhead is determined by the header, plus the flags to delimit the cells or an additional length field in the header. Let Hv = additional overhead octets required to enable the use of variable-length cells. Derive an expression for N in terms of X, H, and Hv.

c. Let $L = 48$, $H = 5$, and $Hv = 2$. Plot N versus message size for fixed- and variable-length cells. Comment on the results.

11.4 Another key design decision for ATM is the size of the data field for fixed-size cells. Let us consider this decision from the point of view of efficiency and delay.

a. Assume that an extended transmission takes place, so that all cells are completely filled. Derive an expression for the efficiency N as a function of H and L.

b. Packetization delay is the delay introduced into a transmission stream by the need to buffer bits until an entire packet is filled before transmission. Derive an expression for this delay as a function of L and the data rate R of the source.

c Common data rates for voice coding are 32 kbps and 64 kbps. Plot packetization delay as a function of L for these two data rates; use a left-hand y axis with a maximum value of 2 ms. On the same graph, plot transmission efficiency as a function of L; use a right-hand y axis with a maximum value of 100%. Comment on the results.

11.5 Suppose that AAL 5 is being used and that the receiver is in an idle state (no incoming cells). Then, a block of user data is transmitted as a sequence of SAR-PDUs.

a. Suppose that a single bit error in one of the SAR-PDUs occurs. What happens at the receiving end?

b. Suppose that one of the cells with AAU = 0 is lost. What happens at the receiving end?

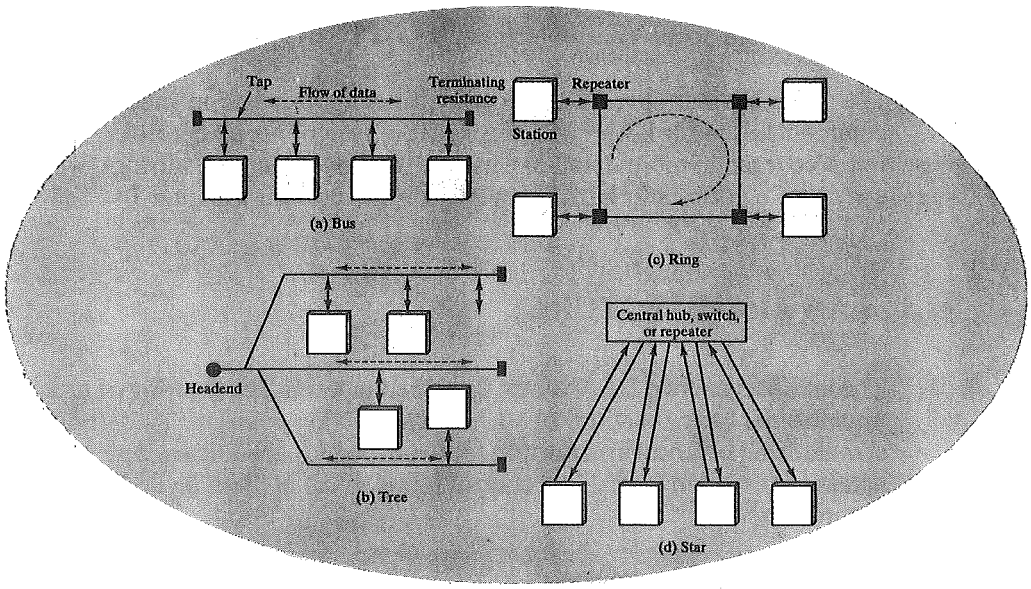
c. Suppose that one of the cells with AAU = 1 is lost. What happens at the receiving end?

11.6 Compare Sustainable Cell Rate and Burst Tolerance, as used in ATM networks, with Committed Information Rate and Excess Burst Size, as used in frame relay networks. Do the respective terms represent the same concepts?

PART THREE Local Area Networks

CHAPTER 12

LAN TECHNOLOGY



In this part, we examine local area networks (LANs) and metropolitan area networks (MANs). These networks share the characteristic of being packet broadcasting networks. With a broadcast communications network, each station is attached to a transmission medium shared by other stations. In its simplest form, a transmission from any one station is broadcast to and received by all other stations. As with packet-switched networks, transmission on a packet broadcasting network is in the form of packets. Table 12.1 provides useful definitions of LANs and MANs, taken from one of the IEEE 802 standards documents.

This chapter begins our discussion of LANs[1] with a description of the protocol architecture that is in common use for implementing LANs. This architecture is also the basis of standardization efforts. Our overview covers the physical, medium access control (MAC), and logical link control (LLC) levels.

Following this overview, the chapter focuses on aspects of LAN technology. The key technology ingredients that determine the nature of a LAN or MAN are

- Topology
- Transmission medium
- Medium access control technique

This chapter surveys the topologies and transmission media that are most commonly used for LANs and MANs. The issue of access control is briefly raised, but is covered in more detail in Chapter 13. The concept of a bridge, which plays a critical role in extending LAN coverage, is discussed in Chapter 14.

12.1 LAN ARCHITECTURE

The architecture of a LAN is best described in terms of a layering of protocols that organize the basic functions of a LAN. This section opens with a description of the standardized protocol architecture for LANs, which encompasses physical, medium access control, and logical link control layers. Each of these layers is then examined in turn.

Protocol Architecture

Protocols defined specifically for LAN and MAN transmission address issues relating to the transmission of blocks of data over the network. In OSI terms, higher-layer protocols (layer 3 or 4 and above) are independent of network architecture and are applicable to LANs, MANs, and WANs. Thus, a discussion of LAN protocols is concerned principally with lower layers of the OSI model.

Figure 12.1 relates the LAN protocols to the OSI architecture (first introduced in Figure 1.10). This architecture was developed by the IEEE 802 committee and has been adopted by all organizations working on the specification of LAN standards. It is generally referred to as the IEEE 802 reference model.

[1] For the sake of brevity, the book often uses LAN when referring to LAN and MAN concerns. The context should clarify when only LAN or both LAN and MAN is meant.

TABLE 12.1 Definitions of LANs and MANs.*

The LANs described herein are distinguished from other types of data networks in that they are optimized for a moderate size geographic area such as a single office building, a warehouse, or a campus. The IEEE 802 LAN is a shared medium peer-to-peer communications network that broadcasts information for all stations to receive. As a consequence, it does not inherently provide privacy. The LAN enables stations to communicate directly using a common physical medium on a point-to-point basis without any intermediate switching node being required. There is always need for an access sublayer in order to arbitrate the access to the shared medium. The network is generally owned, used, and operated by a single organization. This is in contrast to Wide Area Networks (WANs) that interconnect communication facilities in different parts of a country or are used as a public utility. These LANs are also different from networks, such as backplane buses, that are optimized for the interconnection of devices on a desk top or components within a single piece of equipment.

A MAN is optimized for a larger geographical area than a LAN, ranging from several blocks of buildings to entire cities. As with local networks, MANs can also depend on communications channels of moderate-to-high data rates. Error rates and delay may be slightly higher than might be obtained on a LAN. A MAN might be owned and operated by a single organization, but usually will be used by many individuals and organizations. MANs might also be owned and operated as public utilities. They will often provide means for internetworking of local networks. Although not a requirement for all LANs, the capability to perform local networking of integrated voice and data (IVD) devices is considered an optional function for a LAN. Likewise, such capabilities in a network covering a metropolitan area are optional functions of a MAN.

* From IEEE 802 Standard, *Local and Metropolitan Area Networks: Overview and Architecture*, 1990.

Working from the bottom up, the lowest layer of the IEEE 802 reference model corresponds to the *physical layer* of the OSI model, and includes such functions as

- Encoding/decoding of signals
- Preamble generation/removal (for synchronization)
- Bit transmission/reception

In addition, the physical layer of the 802 model includes a specification of the transmission medium and the topology. Generally, this is considered *below* the lowest layer of the OSI model. However, the choice of transmission medium and topology is critical in LAN design, and so a specification of the medium is included.

Above the physical layer are the functions associated with providing service to LAN users. These include

- On transmission, assemble data into a frame with address and error-detection fields.
- On reception, disassemble frame, perform address recognition and error detection.
- Govern access to the LAN transmission medium.
- Provide an interface to higher layers and perform flow and error control.

These are functions typically associated with OSI layer 2. The set of functions in the last bulleted item are grouped into a *logical link control* (LLC) layer. The

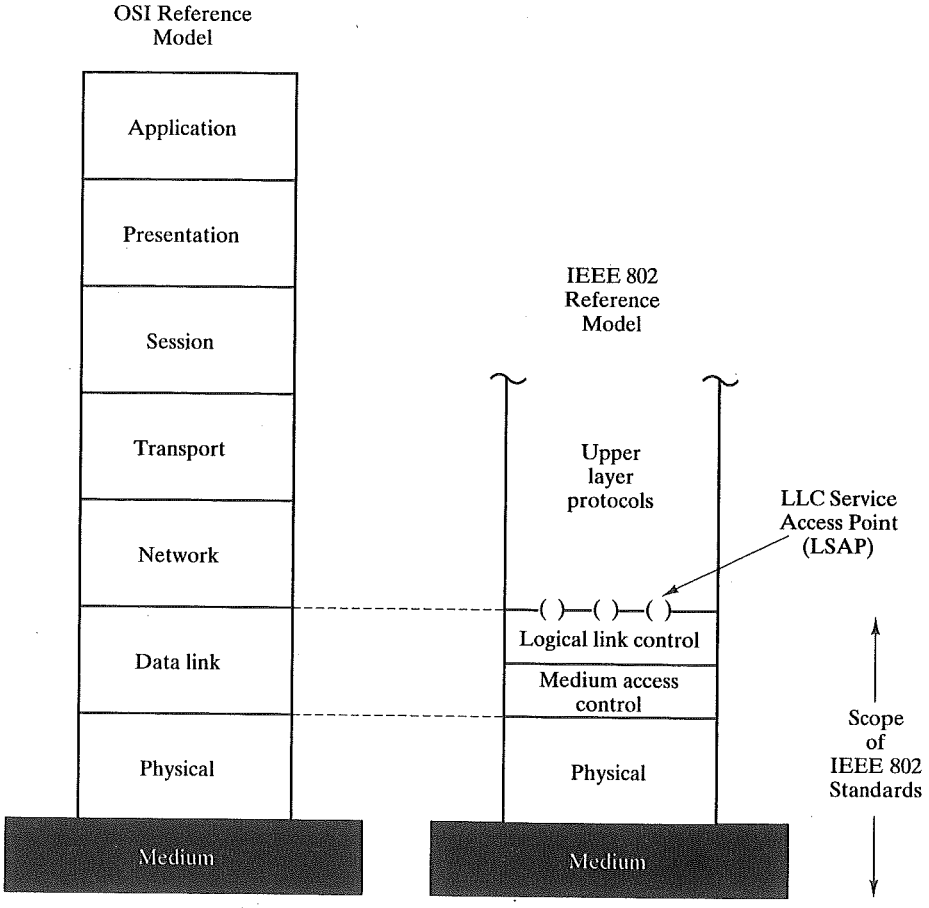


FIGURE 12.1 IEEE 802 protocol layers compared to OSI model.

functions in the first three bullet items are treated as a separate layer, called *medium access control* (MAC). The separation is done for the following reasons:

- The logic required to manage access to a shared-access medium is not found in traditional layer-2 data link control.
- For the same LLC, several MAC options may be provided.

The standards that have been issued are illustrated in Figure 12.2. Most of the standards were developed by a committee known as IEEE 802, sponsored by the Institute for Electrical and Electronics Engineers. All of these standards have subsequently been adopted as international standards by the International Organization for Standardization (ISO).

Figure 12.3 illustrates the relationship between the levels of the architecture (compare Figure 9.17). User data are passed down to LLC, which appends control

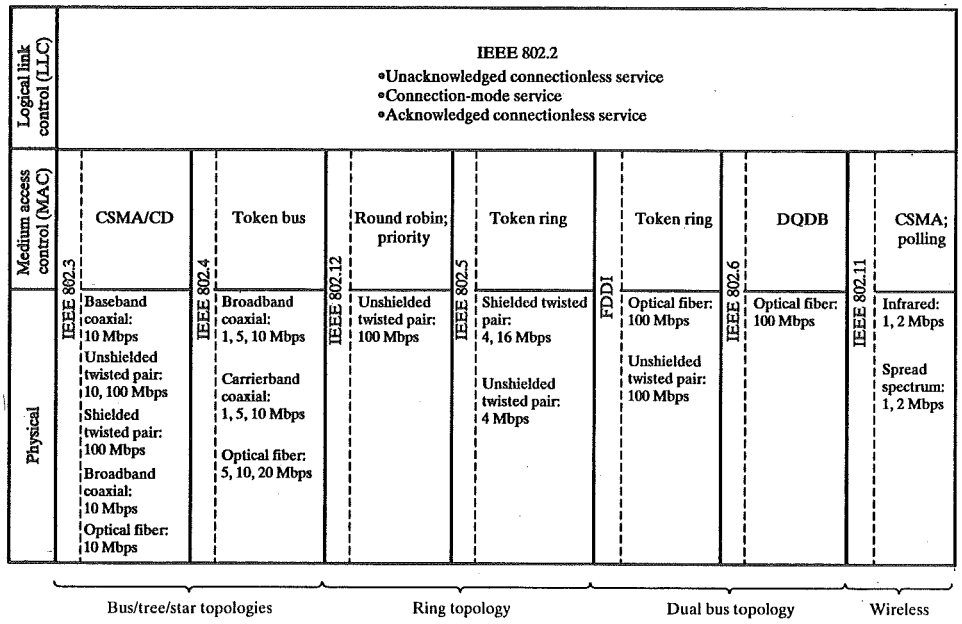


FIGURE 12.2 LAN/MAN standards.

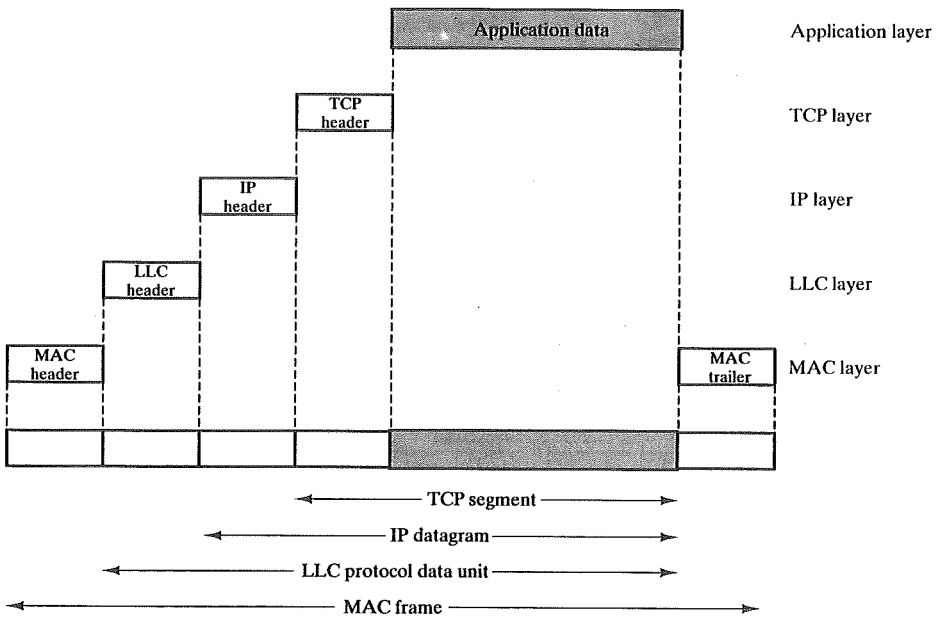


FIGURE 12.3 LAN protocols in context.

information as a header, creating an LLC *protocol data unit* (PDU). This control information is used in the operation of the LLC protocol. The entire LLC PDU is then passed down to the MAC layer, which appends control information at the front and back of the packet, forming a MAC *frame*. Again, the control information in the frame is needed for the operation of the MAC protocol. For context, the figure also shows the use of TCP/IP and an application layer above the LAN protocols.

Topologies

For the physical layer, we confine our discussion for now to an introduction of the basic LAN topologies. The common topologies for LANs are bus, tree, ring, and star (Figure 12.4). The bus is a special case of the tree, with only one trunk and no branches; we shall use the term **bus/tree** when the distinction is unimportant.

Bus and Tree Topologies

Both bus and tree topologies are characterized by the use of a multipoint medium. For the bus, all stations attach, through appropriate hardware interfacing known as a tap, directly to a linear transmission medium, or bus. Full-duplex operation between the station and the tap allows data to be transmitted onto the bus and received from the bus. A transmission from any station propagates the length of the medium in both directions and can be received by all other stations. At each end of the bus is a terminator, which absorbs any signal, removing it from the bus.

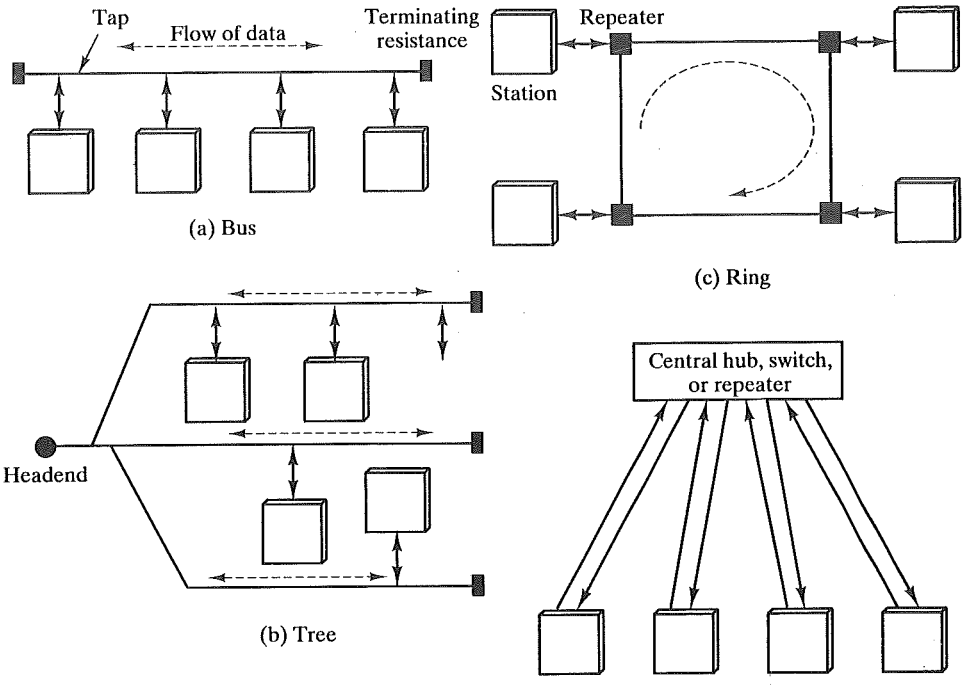


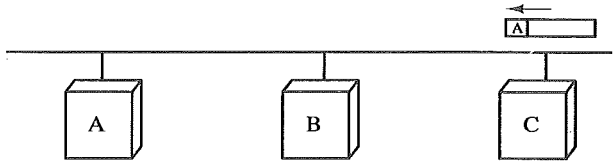
FIGURE 12.4 LAN/MAN topologies.

The tree topology is a generalization of the bus topology. The transmission medium is a branching cable with no closed loops. The tree layout begins at a point known as the *headend*, where one or more cables start, and each of these may have branches. The branches in turn may have additional branches to allow quite complex layouts. Again, a transmission from any station propagates throughout the medium and can be received by all other stations.

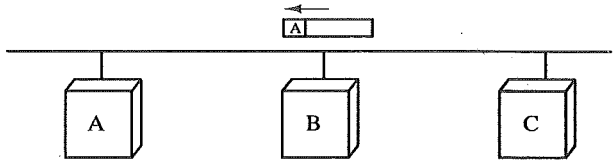
Two problems present themselves in this arrangement. First, because a transmission from any one station can be received by all other stations, there needs to be some way of indicating for whom the transmission is intended. Second, a mechanism is needed to regulate transmission. To see the reason for this, consider that if two stations on the bus attempt to transmit at the same time, their signals will overlap and become garbled. Or, consider that one station decides to transmit continuously for a long period of time.

To solve these problems, stations transmit data in small blocks, known as frames. Each frame consists of a portion of the data that a station wishes to transmit, plus a frame header that contains control information. Each station on the bus is assigned a unique address, or identifier, and the destination address for a frame is included in its header.

Figure 12.5 illustrates the scheme. In this example, station C wishes to transmit a frame of data to A. The frame header includes A's address. As the frame propagates along the bus, it passes B, which observes the address and ignores the frame. A, on the other hand, sees that the frame is addressed to itself and therefore copies the data from the frame as it goes by.



(a) C transmits frame addressed to A



(b) Frame is not addressed to B; B ignores it

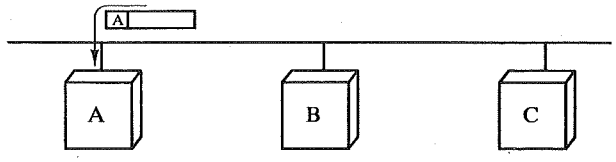


FIGURE 12.5 Frame transmission on a bus LAN.

So the frame structure solves the first problem mentioned above: It provides a mechanism for indicating the intended recipient of data. It also provides the basic tool for solving the second problem, the regulation of access. In particular, the stations take turns sending frames in some cooperative fashion; this involves putting additional control information into the frame header.

With the bus or tree, no special action needs to be taken to remove frames from the medium. When a signal reaches the end of the medium, it is absorbed by the terminator.

Ring Topology

In the ring topology, the network consists of a set of *repeaters* joined by point-to-point links in a closed loop. The repeater is a comparatively simple device, capable of receiving data on one link and transmitting them, bit by bit, on the other link as fast as they are received, with no buffering at the repeater. The links are unidirectional; that is, data are transmitted in one direction only and all are oriented in the same way. Thus, data circulate around the ring in one direction (clockwise or counterclockwise).

Each station attaches to the network at a repeater and can transmit data onto the network through that repeater.

As with the bus and tree, data are transmitted in frames. As a frame circulates past all the other stations, the destination station recognizes its address and copies the frame into a local buffer as it goes by. The frame continues to circulate until it returns to the source station, where it is removed (Figure 12.6).

Because multiple stations share the ring, medium access control is needed to determine at what time each station may insert frames.

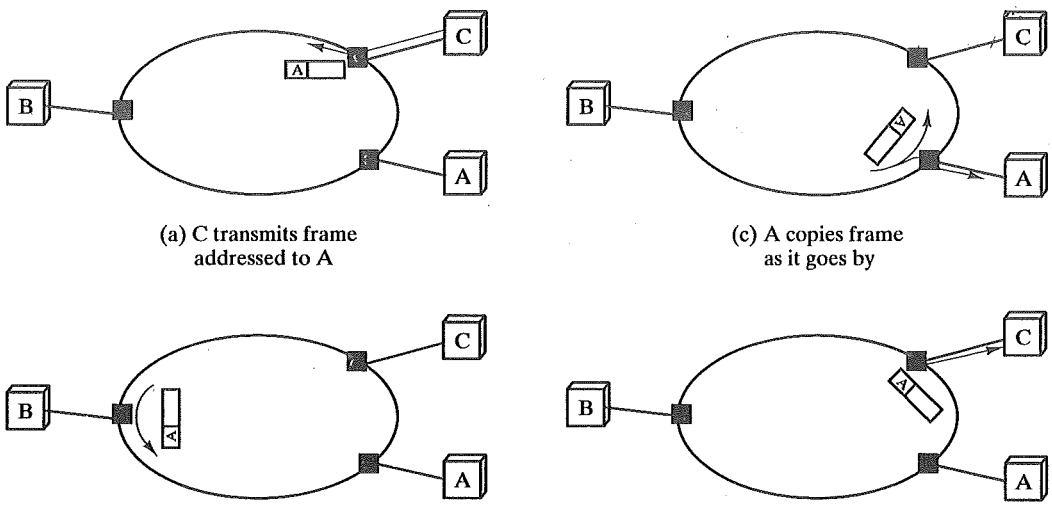


FIGURE 12.6 Frame transmission on a ring LAN.

Star Topology

In the star LAN topology, each station is directly connected to a common central node. Typically, each station attaches to a central node, referred to as the star coupler, via two point-to-point links, one for transmission in each direction.

In general, there are two alternatives for the operation of the central node. One approach is for the central node to operate in a broadcast fashion. A transmission of a frame from one station to the node is retransmitted on all of the outgoing links. In this case, although the arrangement is physically a star, it is logically a bus; a transmission from any station is received by all other stations, and only one station at a time may successfully transmit.

Another approach is for the central node to act as a frame switching device. An incoming frame is buffered in the node and then retransmitted on an outgoing link to the destination station.

Medium Access Control

All LANs and MANs consist of collections of devices that must share the network's transmission capacity. Some means of controlling access to the transmission medium is needed to provide for an orderly and efficient use of that capacity. This is the function of a medium access control (MAC) protocol.

The key parameters in any medium access control technique are where and how. *Where* refers to whether control is exercised in a centralized or distributed fashion. In a centralized scheme, a controller is designated that has the authority to grant access to the network. A station wishing to transmit must wait until it receives permission from the controller. In a decentralized network, the stations collectively perform a medium access control function to dynamically determine the order in which stations transmit. A centralized scheme has certain advantages, such as the following:

- It may afford greater control over access for providing such things as priorities, overrides, and guaranteed capacity.
- It enables the use of relatively simple access logic at each station.
- It avoids problems of distributed coordination among peer entities.

The principal disadvantages of centralized schemes are

- It creates a single point of failure; that is, there is a point in the network that, if it fails, causes the entire network to fail.
- It may act as a bottleneck, reducing performance.

The pros and cons of distributed schemes are mirror images of the points made above.

The second parameter, *how*, is constrained by the topology and is a trade-off among competing factors, including cost, performance, and complexity. In general, we can categorize access control techniques as being either synchronous or asynchronous. With synchronous techniques, a specific capacity is dedicated to a connection; this is the same approach used in circuit switching, frequency-division mul-

tiplexing (FDM), and synchronous time-division multiplexing (TDM). Such techniques are generally not optimal in LANs and MANs because the needs of the stations are unpredictable. It is preferable to be able to allocate capacity in an asynchronous (dynamic) fashion, more or less in response to immediate demand. The asynchronous approach can be further subdivided into three categories: *round robin, reservation, and contention.*

Round Robin

With round robin, each station in turn is given the opportunity to transmit. During that opportunity, the station may decline to transmit or may transmit subject to a specified upper bound, usually expressed as a maximum amount of data transmitted or time for this opportunity. In any case, the station, when it is finished, relinquishes its turn, and the right to transmit passes to the next station in logical sequence. Control of sequence may be centralized or distributed. Polling is an example of a centralized technique.

When many stations have data to transmit over an extended period of time, round robin techniques can be very efficient. If only a few stations have data to transmit over an extended period of time, then there is a considerable overhead in passing the turn from station to station, as most of the stations will not transmit but simply pass their turns. Under such circumstances, other techniques may be preferable, largely depending on whether the data traffic has a stream or bursty characteristic. Stream traffic is characterized by lengthy and fairly continuous transmissions; examples are voice communication, telemetry, and bulk file transfer. Bursty traffic is characterized by short, sporadic transmissions; interactive terminal-host traffic fits this description.

Reservation

For stream traffic, reservation techniques are well suited. In general, for these techniques, time on the medium is divided into slots, much as with synchronous TDM. A station wishing to transmit reserves future slots for an extended or even an indefinite period. Again, reservations may be made in a centralized or distributed fashion.

Contention

For bursty traffic, contention techniques are usually appropriate. With these techniques, no control is exercised to determine whose turn it is; all stations contend for time in a way that can be, as we shall see, rather rough and tumble. These techniques are, of necessity, distributed by nature. Their principal advantage is that they are simple to implement and, under light to moderate load, efficient. For some of these techniques, however, performance tends to collapse under heavy load.

Although both centralized and distributed reservation techniques have been implemented in some LAN products, round robin and contention techniques are the most common.

The discussion above has been somewhat abstract and should become clearer as specific techniques are discussed in Chapter 13. For future reference, Table 12.2 lists the MAC protocols that are defined in LAN and MAN standards.

TABLE 12.2 Standardized medium access control techniques.

	Bus topology	Ring topology	Switched topology
Round robin	Token Bus (IEEE 802.4) Polling (IEEE 802.11)	Token Ring (IEEE 802.5; FDDI)	Request/priority (IEEE 802.12)
Reservation	DQDB (IEEE 802.6)		
Contention	CSMA/CD (IEEE 802.3) CSMA (IEEE 802.11)		CSMA/CD (IEEE 802.3)

MAC Frame Format

The MAC layer receives a block of data from the LLC layer and is responsible for performing functions related to medium access and for transmitting the data. As with other protocol layers, MAC implements these functions, making use of a protocol data unit at its layer; in this case, the PDU is referred to as a MAC frame.

The exact format of the MAC frame differs somewhat for the various MAC protocols in use. In general, all of the MAC frames have a format similar to that of Figure 12.7. The fields of this frame are

- **MAC control.** This field contains any protocol control information needed for the functioning of the MAC protocol. For example, a priority level could be indicated here.
- **Destination MAC address.** The destination physical attachment point on the LAN for this frame.
- **Source MAC address.** The source physical attachment point on the LAN for this frame.

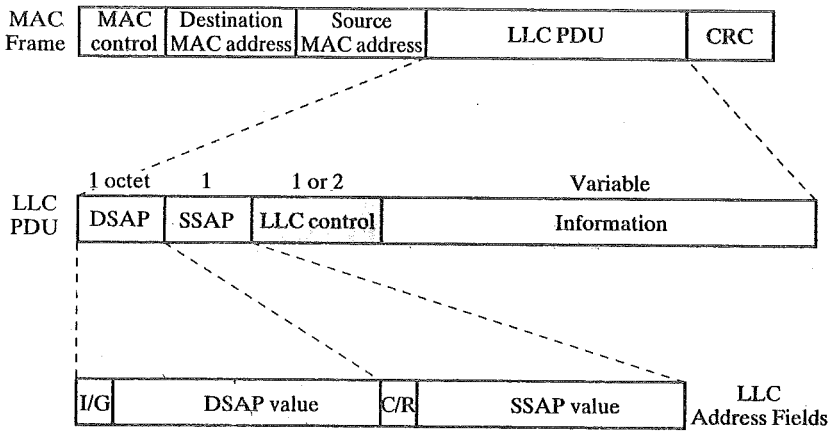


FIGURE 12.7 LLC PDU with generic MAC frame format.

- **LLC.** The LLC data from the next higher layer.
- **CRC.** The cyclic redundancy check field (also known as the frame check sequence, FCS, field). This is an error-detecting code, as we have seen in HDLC and other data link control protocols (Chapter 6).

In most data link control protocols, the data link protocol entity is responsible not only for detecting errors using the CRC, but for recovering from those errors by retransmitting damaged frames. In the LAN protocol architecture, these two functions are split between the MAC and LLC layers. The MAC layer is responsible for detecting errors and discarding any frames that are in error. The LLC layer optionally keeps track of which frames have been successfully received and retransmits unsuccessful frames.

Logical Link Control

The LLC layer for LANs is similar in many respects to other link layers in common use. Like all link layers, LLC is concerned with the transmission of a link-level protocol data unit (PDU) between two stations, without the necessity of an intermediate switching node. LLC has two characteristics not shared by most other link control protocols:

1. It must support the multi-access, shared-medium nature of the link. (This differs from a multidrop line in that there is no primary node.)
2. It is relieved of some details of link access by the MAC layer.

Addressing in LLC involves specifying the source and destination LLC users. Typically, a user is a higher-layer protocol or a network management function in the station. These LLC user addresses are referred to as *service access points* (SAPs), in keeping with OSI terminology for the user of a protocol layer.

We look first at the services that LLC provides to a higher-level user, then at the LLC protocol.

LLC Services

LLC specifies the mechanisms for addressing stations across the medium and for controlling the exchange of data between two users. The operation and format of this standard is based on HDLC. Three services are provided as alternatives for attached devices using LLC:

- **Unacknowledged connectionless service.** This service is a datagram-style service. It is a very simple service that does not involve any of the flow- and error-control mechanisms. Thus, the delivery of data is not guaranteed. However, in most devices, there will be some higher layer of software that deals with reliability issues.
- **Connection-mode service.** This service is similar to that offered by HDLC. A logical connection is set up between two users exchanging data, and flow control and error control are provided.

- **Acknowledged connectionless service.** This is a cross between the previous two services. It provides that datagrams are to be acknowledged, but no prior logical connection is set up.

Typically, a vendor will provide these services as options that the customer can select when purchasing the equipment. Alternatively, the customer can purchase equipment that provides two or all three services and select a specific service based on application.

The *unacknowledged connectionless service* requires minimum logic and is useful in two contexts. First, it will often be the case that higher layers of software will provide the necessary reliability and flow-control mechanism, and it is efficient to avoid duplicating them. For example, either TCP or the ISO transport protocol standard would provide the mechanisms needed to ensure that data are delivered reliably. Second, there are instances in which the overhead of connection establishment and maintenance is unjustified or even counterproductive: for example, data collection activities that involve the periodic sampling of data sources, such as sensors and automatic self-test reports from security equipment or network components. In a monitoring application, the loss of an occasional data unit would not cause distress, as the next report should arrive shortly. Thus, in most cases, the unacknowledged connectionless service is the preferred option.

The *connection-mode service* could be used in very simple devices, such as terminal controllers, that have little software operating above this level. In these cases, it would provide the flow control and reliability mechanisms normally implemented at higher layers of the communications software.

The *acknowledged connectionless service* is useful in several contexts. With the connection-mode service, the logical link control software must maintain some sort of table for each active connection, so as to keep track of the status of that connection. If the user needs guaranteed delivery, but there are a large number of destinations for data, then the connection-mode service may be impractical because of the large number of tables required; an example is a process-control or automated factory environment where a central site may need to communicate with a large number of processors and programmable controllers; another use is the handling of important and time-critical alarm or emergency control signals in a factory. Because of their importance, an acknowledgment is needed so that the sender can be assured that the signal got through. Because of the urgency of the signal, the user might not want to take the time to first establish a logical connection and then send the data.

LLC Protocol

The basic LLC protocol is modeled after HDLC, and has similar functions and formats. The differences between the two protocols can be summarized as follows:

1. LLC makes use of the asynchronous, balanced mode of operation of HDLC in order to support connection-mode LLC service; this is referred to as type 2 operation. The other HDLC modes are not employed.
2. LLC supports a connectionless service using the unnumbered information PDU; this is known as type 1 operation.

3. LLC supports an acknowledged connectionless service by using two new unnumbered PDUs; this is known as type 3 operation.
4. LLC permits multiplexing by the use of LLC service access points (LSAPs).

All three LLC protocols employ the same PDU format (Figure 12.7), which consists of four fields. The DSAP and SSAP fields each contain 7-bit addresses, which specify the destination and source users of LLC. One bit of the DSAP indicates whether the DSAP is an individual or group address. One bit of the SSAP indicates whether the PDU is a command or response PDU. The format of the LLC control field is identical to that of HDLC (Figure 6.10), using extended (7-bit) sequence numbers.

For *type 1 operation*, which supports the unacknowledged connectionless service, the unnumbered information (UI) PDU is used to transfer user data. There is no acknowledgment, flow control, or error control. However, there is error detection and discard at the MAC level.

Two other PDUs are used to support management functions associated with all three types of operation. Both PDUs are used in the following fashion. An LLC entity may issue a command (C/R bit = 0) XID or TEST. The receiving LLC entity issues a corresponding XID or TEST in response. The XID PDU is used to exchange two types of information: types of operation supported and window size. The TEST PDU is used to conduct a loop-back test of the transmission path between two LLC entities. Upon receipt of a TEST command PDU, the addressed LLC entity issues a TEST response PDU as soon as possible.

With *type 2 operation*, a data link connection is established between two LLC SAPs prior to data exchange. Connection establishment is attempted by the type 2 protocol in response to a request from a user. The LLC entity issues a SABME PDU[2] to request a logical connection with the other LLC entity. If the connection is accepted by the LLC user designated by the DSAP, then the destination LLC entity returns an unnumbered acknowledgment (UA) PDU. The connection is henceforth uniquely identified by the pair of user SAPs. If the destination LLC user rejects the connection request, its LLC entity returns a disconnected mode (DM) PDU.

Once the connection is established, data are exchanged using information PDUs, as in HDLC. The information PDUs include send and receive sequence numbers, for sequencing and flow control. The supervisory PDUs are used, as in HDLC, for flow control and error control. Either LLC entity can terminate a logical LLC connection by issuing a disconnect (DISC) PDU.

With *type 3 operation*, each transmitted PDU is acknowledged. A new (not found in HDLC) unnumbered PDU, the Acknowledged Connectionless (AC) Information PDU is defined. User data are sent in AC command PDUs and must be acknowledged using an AC response PDU. To guard against lost PDUs, a 1-bit sequence number is used. The sender alternates the use of 0 and 1 in its AC com-

[2] This stands for Set Asynchronous Balanced Mode Extended. It is used in HDLC to choose ABM and to select extended sequence numbers of seven bits. Both ABM and 7-bit sequence numbers are mandatory in type 2 operation.

mand PDU, and the receiver responds with an AC PDU with the opposite number of the corresponding command. Only one PDU in each direction may be outstanding at any time.

12.2 BUS/TREE LANs

This section provides some technical details on bus/tree topology LANs and MANs. The section begins with an overview of the general characteristics of this topology. The remainder of the section examines the use of coaxial cable and optical fiber for implementing this topology.

Characteristics of the Bus/Tree Topology

The bus/tree topology is a multipoint configuration. That is, there are more than two devices connected to the medium and capable of transmitting on the medium. This situation gives rise to several design issues, the first of which is the need for a medium access control technique.

Another design issue has to do with signal balancing. When two stations exchange data over a link, the signal strength of the transmitter must be adjusted to be within certain limits. The signal must be strong enough so that, after attenuation across the medium, it meets the receiver's minimum signal-strength requirements. It must also be strong enough to maintain an adequate signal-to-noise ratio. On the other hand, the signal must not be so strong that it overloads the circuitry of the transmitter, as the signal would become distorted. Although easily accomplished for a point-to-point link, signal balancing is no easy task for a multipoint line. If any station can transmit to any other station, then the signal balancing must be performed for all permutations of stations taken two at a time. For n stations, that works out to $n \times (n - 1)$ permutations. So, for a 200-station network (not a particularly large system), 39,800 signal-strength constraints must be satisfied simultaneously; with interdevice distances ranging from tens to thousands of meters, this would be an extremely difficult task for any but small networks. In systems that use radio-frequency (RF) signals, the problem is compounded because of the possibility of RF signal interference across frequencies. A common solution is to divide the medium into smaller segments within which pairwise balancing is possible, using amplifiers or repeaters between segments.

Baseband Coaxial Cable

For bus/tree LANs, the most popular medium is coaxial cable. The two common transmission techniques that are used on coaxial cable are baseband and broadband, which are compared in Table 12.3. This subsection is devoted to baseband systems, while the next section discusses broadband LANs.

A baseband LAN or MAN is defined as one that uses digital signaling; that is, the binary data to be transmitted are inserted onto the cable as a sequence of voltage pulses, usually using Manchester or Differential Manchester encoding (see

TABLE 12.3 Transmission techniques for coaxial cable bus/tree LANs.

Baseband	Broadband
Digital signaling	Analog signaling (requires RF modem)
Entire bandwidth consumed by signal—no frequency division multiplexing (FDM)	FDM possible—multiple channels for data, video, audio
Bidirectional	Unidirectional
Bus topology	Bus or tree topology
Distance: up to a few kilometers	Distance: up to tens of kilometers

Figure 4.2). The nature of digital signals is such that the entire frequency spectrum of the cable is consumed. Hence, it is not possible to have multiple channels (frequency-division multiplexing) on the cable. Transmission is bidirectional. That is, a signal inserted at any point on the medium propagates in both directions to the ends, where it is absorbed (Figure 12.8a). The digital signaling requires a bus topology; unlike analog signals, digital signals cannot easily be propagated through the branching points required for a tree topology. Baseband bus systems can extend only a few kilometers, at most; this is because the attenuation of the signal, which is most pronounced at higher frequencies, causes a blurring of the pulses and a weakening of the signal to the extent that communication over larger distances is impractical.

The original use of baseband coaxial cable for a bus LAN was the Ethernet system, which operates at 10 Mbps. Ethernet became the basis of the IEEE 802.3 standard.

Most baseband coaxial cable systems use a special 50-ohm cable rather than the standard CATV 75-ohm cable. These values refer to the impedance of the cable. Roughly speaking, impedance is a measure of how much voltage must be applied to the cable to achieve a given signal strength. For digital signals, the 50-ohm cable suffers less intense reflections from the insertion capacitance of the taps and provides better immunity against low-frequency electromagnetic noise, compared to 75-ohm cable.

As with any transmission system, there are engineering trade-offs involving data rate, cable length, number of taps, and the electrical characteristics of the cable and the transmit/receive components. For example, the lower the data rate, the longer the cable can be. That statement is true for the following reason: When a signal is propagated along a transmission medium, the integrity of the signal suffers due to attenuation, noise, and other impairments. The longer the length of propagation, the greater the effect, thereby increasing the probability of error. However, at a lower data rate, the individual pulses of a digital signal last longer and can be recovered in the presence of impairments more easily than higher-rate, shorter pulses.

Here is one example that illustrates some of the trade-offs. The Ethernet specification and the original IEEE 802.3 standard specified the use of 50-ohm cable with a 0.4-inch diameter, and a data rate of 10 Mbps. With these parameters, the maximum length of the cable is set at 500 meters. Stations attach to the cable by

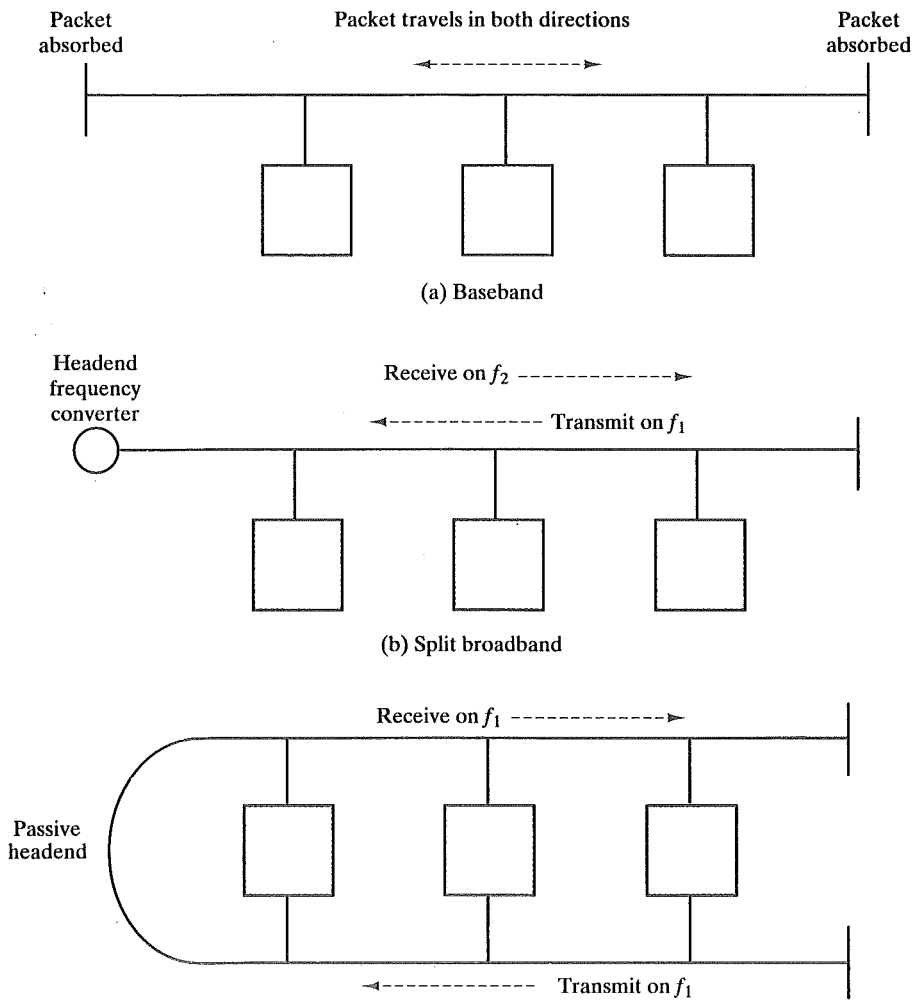


FIGURE 12.8 Baseband and broadband transmission techniques.

means of a tap, with the distance between any two taps being a multiple of 2.5 m; this is to ensure that reflections from adjacent taps do not add in phase [YEN83]. A maximum of 100 taps is allowed. In IEEE jargon, this system is referred to as 10BASE5 (10 Mbps, baseband, 500-m cable length).

To provide a lower-cost system for personal computer LANs, IEEE 802.3 later added a 10BASE2 specification. Table 12.4 compares this scheme, dubbed Cheapernet, with 10BASE5. The key change is the use of a thinner (0.25 in) cable of the type employed in products such as public address systems. The thinner cable is more flexible; thus, it is easier to bend around corners and bring to a workstation rather than installing a cable in the wall and having to provide a drop cable between the main cable and the workstation. The cable is easier to install and uses cheaper electronics than the thicker cable. On the other hand, the thinner cable suffers

TABLE 12.4 IEEE 802.3 specifications for 10-Mbps baseband coaxial cable bus LANs.

	10BASE5	10BASE2
Data rate	10 Mbps	10 Mbps
Maximum Segment Length	500 m	185 m
Network Span	2500 m	1000 m
Nodes per Segment	100	30
Node Spacing	2.5 m	0.5 m
Cable Diameter	0.4 in	0.25 in

greater attenuation and lower noise resistance than the thicker cable; as a result, it supports fewer taps over a shorter distance.

To extend the length of the network, a repeater may be used. This device works in a somewhat different fashion than the repeater on the ring. The bus repeater is not used as a device attachment point and is capable of transmitting in both directions. A repeater joins two segments of cable and passes digital signals in both directions between the two segments. A repeater is transparent to the rest of the system; as it does no buffering, it does not logically isolate one segment from another. So, for example, if two stations on different segments attempt to transmit at the same time, their packets will interfere with each other (collide). To avoid multipath interference, only one path of segments and repeaters is allowed between any two stations. Figure 12.9 illustrates a multiple-segment baseband bus LAN.

Broadband Coaxial Cable

In the local network context, the term broadband refers to coaxial cable on which analog signaling is used. Table 12.3 summarizes the key characteristics of broad-

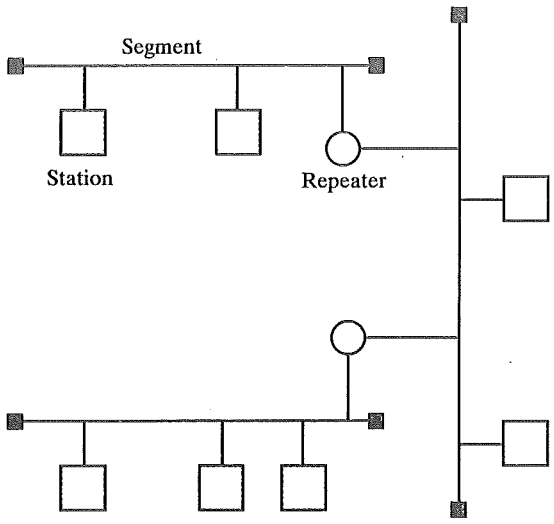


FIGURE 12.9 Baseband configuration.

band systems. As mentioned, broadband implies the use of analog signaling. FDM is possible, as the frequency spectrum of the cable can be divided into channels or sections of bandwidth. Separate channels can support data traffic, video, and radio signals. Broadband components allow splitting and joining operations; hence, both bus and tree topologies are possible. Much greater distances—tens of kilometers—are possible with broadband compared to baseband because the analog signals that carry the digital data can propagate greater distances before the noise and attenuation damage the data.

Dual and Split Configurations

As with baseband, stations on a broadband LAN attach to the cable by means of a tap. Unlike baseband, however, broadband is inherently a unidirectional medium; the taps that are used allow signals inserted onto the medium to propagate in only one direction. The primary reason for this is that it is unfeasible to build amplifiers that will pass signals of one frequency in both directions. This unidimensional property means that only those stations "downstream" from a transmitting station can receive its signals. How, then, to achieve full connectivity?

Clearly, two data paths are needed. These paths are joined at a point on the network known as the *headend*. For a bus topology, the headend is simply one end of the bus. For a tree topology, the headend is the root of the branching tree. All stations transmit on one path toward the headend (inbound). Signals arriving at the headend are then propagated along a second data path away from the headend (outbound). All stations receive on the outbound path.

Physically, two different configurations are used to implement the inbound and outbound paths (Figure 12.8b and c). On a *dual-cable* configuration, the inbound and outbound paths are separate cables, with the headend simply a passive connector between the two. Stations send and receive on the same frequency.

By contrast, on a *split* configuration, the inbound and outbound paths are different frequency bands on the same cable. Bidirectional amplifiers[3] pass lower frequencies inbound, and higher frequencies outbound. Between the inbound and outbound frequency bands is a guardband, which carries no signals and serves merely as a separator. The headend contains a device for converting inbound frequencies to outbound frequencies.

The frequency-conversion device at the headend can be either an analog or digital device. An analog device, known as a *frequency translator*, converts a block of frequencies from one range to another. A digital device, known as a *remodulator*, recovers the digital data from the inbound analog signal and then retransmits the data on the outbound frequency. Thus, a remodulator provides better signal quality by removing all of the accumulated noise and attenuation and transmitting a cleaned-up signal.

Split systems are categorized by the frequency allocation of the two paths, as shown in Table 12.5. Subsplit, commonly used by the cable television industry, was designed for metropolitan area television distribution, with limited subscriber-to-central-office communication. It provides the easiest way to upgrade existing

[3] Unfortunately, this terminology is confusing, as we have said that broadband is inherently a unidirectional medium. At a given frequency, broadband is unidirectional. However, there is no difficulty in having signals in nonoverlapping frequency bands traveling in opposite directions on the cable.

TABLE 12.5 Common broadband cable frequency splits.

Format	Inbound Frequency Band	Outbound Frequency Band	Maximum Two-way Bandwidth
Subsplit	5 to 30 MHz	54 to 400 MHz	25 Mhz
Midsplit	5 to 116 MHz	168 to 400 MHz	111 Mhz
Highsplit	5 to 174 MHz	232 to 400 MHz	168 Mhz
Dual Cable	40 to 400 MHz	40 to 400 MHz	360 Mhz

one-way cable systems to two-way operation. Subsplit has limited usefulness for local area networking because a bandwidth of only 25 MHz is available for two-way communication. Midsplit is more suitable for LANs, because it provides a more equitable distribution of bandwidth. However, midsplit was developed at a time when the practical spectrum of a cable-TV cable was 300 MHz, whereas a spectrum of 400 to 450 MHz is now available. Accordingly, a highsplit specification has been developed to provide greater two-way bandwidth for a split cable system.

The differences between split and dual configurations are minor. The split system is useful when a single cable plant is already installed in a building. If a large amount of bandwidth is needed, or the need is anticipated, then a dual cable system is indicated. Beyond these considerations, it is a matter of a trade-off between cost and size. The single-cable system has the fixed cost of the headend remodulator or frequency translator. The dual cable system makes use of more cable, taps, splitters, and amplifiers. Thus, dual cable is cheaper for smaller systems, where the fixed cost of the headend is noticeable, and single cable is cheaper for larger systems, where incremental costs dominate.

Carrierband

There is another application of analog signaling on a LAN, known as carrierband, or single-channel broadband. In this case, the entire spectrum of the cable is devoted to a single transmission path for the analog signals; no frequency-division multiplexing is possible.

Typically, a carrierband LAN has the following characteristics. Bidirectional transmission, using a bus topology, is employed. Hence, there can be no amplifiers, and there is no need for a headend. Although the entire spectrum is used, most of the signal energy is concentrated at relatively low frequencies, which is an advantage, because attenuation is less at lower frequencies.

Because the cable is dedicated to a single task, it is not necessary to take care that the modem output be confined to a narrow bandwidth. Energy can spread over the entire spectrum. As a result, the electronics are simple and relatively inexpensive. Typically, some form of frequency-shift keying (FSK) is used. Carrierband would appear to give comparable performance, at a comparable price, to baseband.

Optical Fiber Bus

Several approaches can be taken in the design of a fiber bus topology LAN or MAN. The differences have to do with the nature of the taps into the bus and the detailed topology.

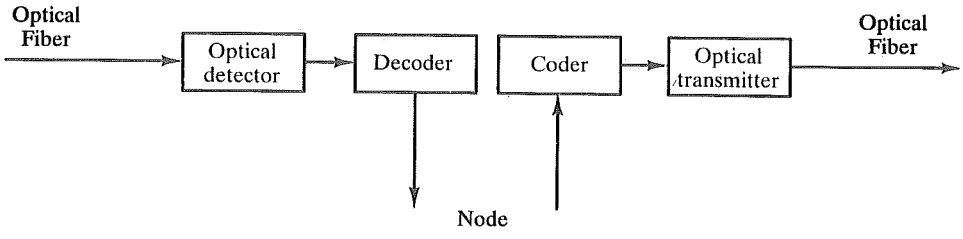
Optical Fiber Taps

With an optical fiber bus, either an active or passive tap can be used. In the case of an active tap (Figure 12.10a), the following steps occur:

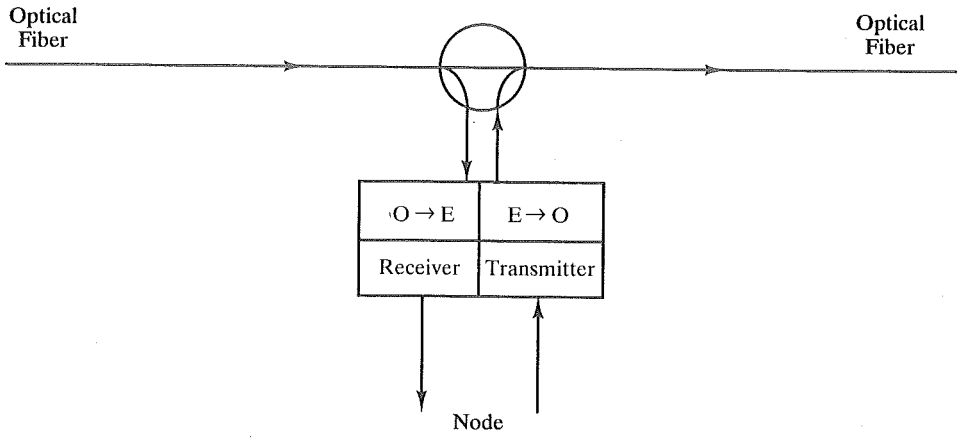
1. Optical signal energy enters the tap from the bus.
2. Clocking information is recovered from the signal, and the signal is converted to an electrical signal.
3. The converted signal is presented to the node and perhaps modified by the latter.
4. The optical output (a light beam) is modulated according to the electrical signal and launched into the bus.

In effect, the bus consists of a chain of point-to-point links, and each node acts as a repeater. Each tap actually consists of two of these active couplers and requires two fibers; this is because of the inherently unidirectional nature of the device of Figure 12.10a.

In the case of a passive tap (Figure 12.10b), the tap extracts a portion of the optical energy from the bus for reception and it injects optical energy directly into the medium for transmission. Thus, there is a single run of cable rather than a chain



(a) Active tap



(b) Passive tap

FIGURE 12.10 Optical fiber bus taps.

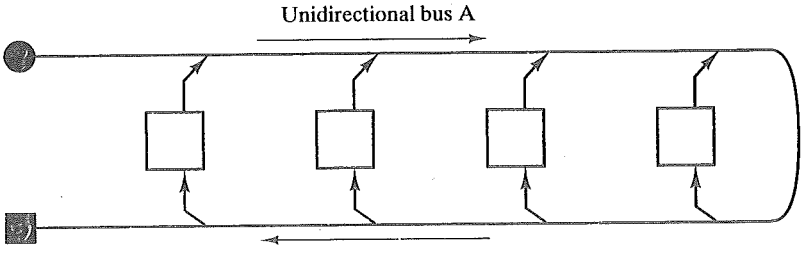
of point-to-point links. This passive approach is equivalent to the type of taps typically used for twisted pair and coaxial cable. Each tap must connect to the bus twice: once for transmit and once for receive.

The electronic complexity and interface cost are drawbacks for the implementation of the active tap. Also, each tap will add some increment of delay, just as in the case of a ring. For passive taps, the lossy nature of pure optical taps limits the number of devices and the length of the medium. However, the performance of such taps has improved sufficiently in recent years so to make fiber bus networks practical.

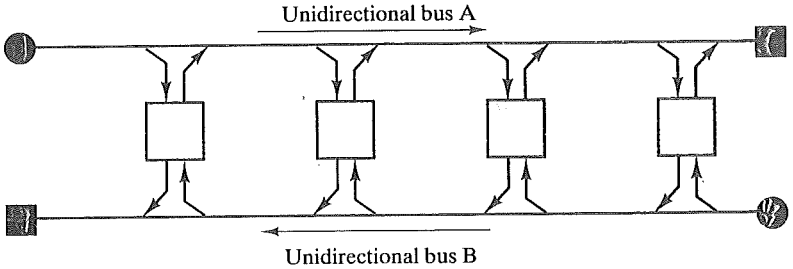
Optical Fiber Bus Configurations

A variety of configurations for the optical fiber bus have been proposed, all of which fall into two categories: those that use a single bus and those that use two buses.

Figure 12.11a shows a typical single-bus configuration, referred to as a loop bus. The operation of this bus is essentially the same as that of the dual-bus broadband coaxial system described earlier. Each station transmits on the bus in the direction toward the headend, and receives on the bus in the direction away from the headend. In addition to the two connections shown, some MAC protocols require that each station have an additional **sense tap** on the inbound (toward the headend) portion of the bus. The sense tap is able to sense the presence or absence of light on the fiber, but it is not able to recover data.



(a) Loop bus



(b) Dual bus

FIGURE 12.11 Optical fiber bus configurations.

Figure 12.11b shows the two-bus configuration. Each station attaches to both buses and has both transmit and receive taps on both buses. On each bus, a station may transmit only to those stations downstream from it. By using both buses, a station may transmit to, and receive from, all other stations. A given node, however, must know which bus to use to transmit to another node; if such information were unavailable, all data would have to be sent out on both buses; this is the configuration used in the IEEE 802.6 MAN, and is described in Chapter 13.

12.3 RING LANs

Characteristics of Ring LANs

A ring consists of a number of repeaters, each connected to two others by unidirectional transmission links to form a single closed path. Data are transferred sequentially, bit by bit, around the ring from one repeater to the next. Each repeater regenerates and retransmits each bit.

For a ring to operate as a communication network, three functions are required: data insertion, data reception, and data removal. These functions are provided by the repeaters. Each repeater, in addition to serving as an active element on the ring, serves as a device attachment point. Data insertion is accomplished by the repeater. Data are transmitted in packets, each of which contains a destination address field. As a packet circulates past a repeater, the address field is copied. If the attached station recognizes the address, the remainder of the packet is copied.

Repeaters perform the data insertion and reception functions in a manner not unlike that of taps, which serve as device attachment points on a bus or tree. Data removal, however, is more difficult on a ring. For a bus or tree, signals inserted onto the line propagate to the endpoints and are absorbed by terminators. Hence, shortly after transmission ceases, the bus or tree is clean of data. However, because the ring is a closed loop, a packet will circulate indefinitely unless it is removed. A packet may by removed by the addressed repeater. Alternatively, each packet could be removed by the transmitting repeater after it has made one trip around the loop. This latter approach is more desirable because (1) it permits automatic acknowledgment and (2) it permits multicast addressing: one packet sent simultaneously to multiple stations.

A variety of strategies can be used for determining how and when packets are inserted onto the ring. These strategies are, in effect, medium access control protocols, and are discussed in Chapter 13.

The repeater, then, can be seen to have two main purposes: (1) to contribute to the proper functioning of the ring by passing on all the data that come its way, and (2) to provide an access point for attached stations to send and receive data. Corresponding to these two purposes are two states (Figure 12.12): the listen state and the transmit state.

In the listen state, each received bit is retransmitted with a small delay, required to allow the repeater to perform required functions. Ideally, the delay should be on the order of one bit time (the time it takes for a repeater to transmit one complete bit onto the outgoing line). These functions are

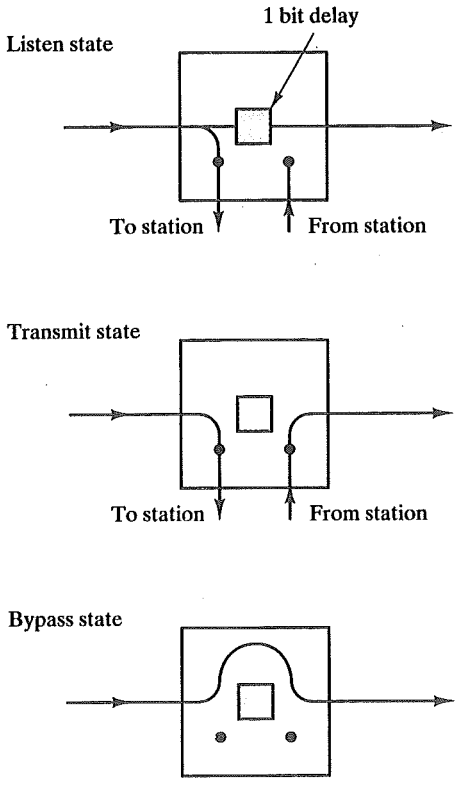


FIGURE 12.12 Ring repeater states.

- Scan passing bit stream for pertinent patterns. Chief among these is the address or addresses of attached stations. Another pattern, used in the token control strategy explained later, indicates permission to transmit. Note that to perform the scanning function, the repeater must have some knowledge of packet format.
- Copy each incoming bit and send it to the attached station, while continuing to retransmit each bit. This will be done for each bit of each packet addressed to this station.
- Modify a bit as it passes by. In certain control strategies, bits may be modified, for example, to indicate that the packet has been copied; this would serve as an acknowledgment.

When a repeater's station has data to send, and when the repeater, based on the control strategy, has permission to send, the repeater enters the transmit state. In this state, the repeater receives bits from the station and retransmits them on its outgoing link. During the period of transmission, bits may appear on the incoming ring link. There are two possibilities, and they are treated differently:

- The bits could be from the same packet that the repeater is still in the process of sending. This will occur if the bit length of the ring is shorter than the packet. In this case, the repeater passes the bits back to the station, which can check them as a form of acknowledgment.
- For some control strategies, more than one packet could be on the ring at the same time. If the repeater, while transmitting, receives bits from a packet it did not originate, it must buffer them to be transmitted later.

These two states, listen and transmit, are sufficient for proper ring operation. A third state, the bypass state, is also useful. In this state, a bypass relay can be activated so that signals propagate past the repeater with no delay other than from medium propagation. The bypass relay affords two benefits: (1) it provides a partial solution to the reliability problem, discussed later, and (2) it improves performance by eliminating repeater delay for those stations that are not active on the network.

Twisted pair, baseband coax, and fiber optic cable can all be used to provide the repeater-to-repeater links. Broadband coax, however, could not easily be used. Each repeater would have to be capable, asynchronously, of receiving and transmitting data on multiple channels.

Timing Jitter

On a ring transmission medium, some form of clocking is included with the signal, as for example with the use of Differential Manchester encoding (Section 4.1). As data circulate around the ring, each repeater receives the data, and recovers the clocking for two purposes: first, to know when to sample the incoming signal to recover the bits of data, and second, to use the clocking for transmitting the signal to the next repeater. This clock recovery will deviate in a random fashion from the mid-bit transitions of the received signal for several reasons, including noise during transmission and imperfections in the receiver circuitry; the predominant reason, however, is delay distortion (described in Section 2.3). The deviation of clock recovery is known as timing jitter.

As each repeater receives incoming data, it issues a clean signal with no distortion. However, the timing error is not eliminated. Thus, the digital pulse width will expand and contract in a random fashion as the signal travels around the ring and the timing jitter accumulates. The cumulative effect of the jitter is to cause the bit latency, or bit length, of the ring to vary. However, unless the latency of the ring remains constant, bits will be dropped (not retransmitted) as the latency of the ring decreases, or they will be added as the latency increases.

This timing jitter places a limitation on the number of repeaters in a ring. Although this limitation cannot be entirely overcome, several measures can be taken to improve matters. In essence, two approaches are used in combination. First, each repeater can include a phase-lock loop. This is a device that uses feedback to minimize the deviation from one bit time to the next. Second, a buffer can be used at one or more repeaters. The buffer is initialized to hold a certain number of bits, and expands and contracts to keep the bit length of the ring constant. The

combination of phase-locked loops and a buffer significantly increases maximum feasible ring size.

Potential Ring Problems

There are a number of potential problems with the ring topology: A break in any link or the failure of a repeater disables the entire network; installation of a new repeater to support new devices requires the identification of two nearby, topologically adjacent repeaters; timing jitter must be dealt with; and finally, because the ring is closed, a means is needed to remove circulating packets, with backup techniques to guard against error.

The last problem is a protocol issue and will be discussed later. The remaining problems can be handled by a refinement of the ring topology and will be discussed next.

The Star-Ring Architecture

Two observations can be made about the basic ring architecture described above. First, there is a practical limit to the number of repeaters on a ring. This limit is suggested by the jitter, reliability, and maintenance problems just cited, and by the accumulating delay of a large number of repeaters. A limit of a few hundred repeaters seems reasonable. Second, the functioning of the ring does not depend on the actual routing of the cables that link the repeaters.

These observations have led to the development of a refined ring architecture, the star-ring, which overcomes some of the problems of the ring and allows the construction of larger local networks.

As a first step, consider the rearrangement of a ring into a star. This is achieved by having the interrepeater links all thread through a single site. This ring wiring concentrator has a number of advantages. Because there is centralized access to the signal on every link, it is a simple matter to isolate a fault. A message can be launched into the ring and tracked to see how far it gets without mishap. A faulty segment can be disconnected and repaired at a later time. New repeaters can easily be added to the ring: Simply run two cables from the new repeater to the site of the ring wiring concentration and splice into the ring.

The bypass relay associated with each repeater can be moved into the ring wiring concentrator. The relay can automatically bypass its repeater and two links in the event of any malfunction. A nice effect of this feature is that the transmission path from one working repeater to the next is approximately constant; thus, the range of signal levels to which the transmission system must automatically adapt is much smaller.

The ring wiring concentrator permits rapid recovery from a cable or repeater failure. Nevertheless, a single failure could, at least temporarily, disable the entire network. Furthermore, throughput and jitter considerations still place a practical upper limit on the number of stations in a ring, as each repeater adds an increment of delay. Finally, in a spread-out network, a single wire concentration site dictates a great deal of cable.

To attack these remaining problems, a LAN consisting of multiple rings connected by bridges can be constructed. We explore the use of bridges in Chapter 14.

Bus Versus Ring

For the user with a large number of devices and high-capacity requirements, the bus or tree broadband LAN seems the best suited to the requirements. For more moderate requirements, however, the choice between a baseband bus LAN and a ring LAN is not at all clear-cut.

The baseband bus is the simpler system. Passive taps rather than active repeaters are used. There is no need for the complexity of bridges and ring wiring concentrators.

The most important benefit of the ring is that it uses point-to-point communication links, and here there are a number of implications. First, because the transmitted signal is regenerated at each node, transmission errors are minimized and greater distances can be covered than with baseband bus. Broadband bus/tree can cover a similar range, but cascaded amplifiers can result in loss of data integrity at high data rates. Second, the ring can accommodate optical fiber links, which provide very high data rates and excellent electromagnetic interference (EMI) characteristics. Finally, the electronics and maintenance of point-to-point lines are simpler than for multipoint lines.

12.4 STAR LANs

Twisted Pair Star LANs

In recent years, there has been increasing interest in the use of twisted pair as a transmission medium for LANs. From the earliest days of commercial LAN availability, twisted pair bus LANs have been popular. However, such LANs suffer in comparison with a coaxial cable LAN. First of all, the apparent cost advantage of twisted pair is not as great as it might seem, at least when a linear bus layout is used. True, twisted pair cable is less expensive than coaxial cable. On the other hand, much of the cost of LAN wiring is in the labor cost of installing the cable, which is no greater for coaxial cable than for twisted pair. Secondly, coaxial cable provides superior signal quality, and therefore can support more devices over longer distances at higher data rates than twisted pair.

The renewed interest in twisted pair, at least in the context of bus/tree type LANs, is in the use of unshielded twisted pair in a star-wiring arrangement. The reason for the interest is that unshielded twisted pair is simply telephone wire, and virtually all office buildings are equipped with spare twisted pairs running from wiring closets to each office. This yields several benefits when deploying a LAN:

1. There is essentially no installation cost with unshielded twisted pair, as the wire is already there. Coaxial cable has to be pulled. In older buildings, this may be difficult because existing conduits may be crowded.
2. In most office buildings, it is impossible to anticipate all the locations where network access will be needed. Because it is extravagantly expensive to run coaxial cable to every office, a coaxial cable-based LAN will typically cover only a portion of a building. If equipment subsequently has to be moved to an

office not covered by the LAN, significant expense is involved in extending the LAN coverage. With telephone wire, this problem does not arise, as all offices are covered.

The most popular approach to the use of unshielded twisted pair for a LAN is therefore a star-wiring approach. The products on the market use a scheme suggested by Figure 12.13, in which the central element of the star is an active element, referred to as the *hub*. Each station is connected to the hub by two twisted pairs (transmit and receive). The hub acts as a repeater: When a single station transmits, the hub repeats the signal on the outgoing line to each station.

Note that although this scheme is physically a star, it is logically a bus: A transmission from any one station is received by all other stations, and, if two stations transmit at the same time, there will be a collision.

Multiple levels of hubs can be cascaded in a hierarchical configuration. Figure 12.14 illustrates a two-level configuration. There is one *header hub* (HHUB) and one or more *intermediate hubs* (IHUB). Each hub may have a mixture of stations and other hubs attached to it from below. This layout fits well with building wiring practices. Typically, there is a wiring closet on each floor of an office building, and a hub can be placed in each one. Each hub could service the stations on its floor.

Figure 12.15 shows an abstract representation of the intermediate and header hubs. The header hub performs all the functions described previously for a single-hub configuration. In the case of an intermediate hub, any incoming signal from below is repeated upward to the next higher level. Any signal from above is repeated on all lower-level outgoing lines. Thus, the logical bus characteristic is retained: A transmission from any one station is received by all other stations, and, if two stations transmit at the same time, there will be a collision.

Optical Fiber Star

One of the first commercially available approaches for fiber LANs was the passive star coupler. The passive star coupler is fabricated by fusing together a number of

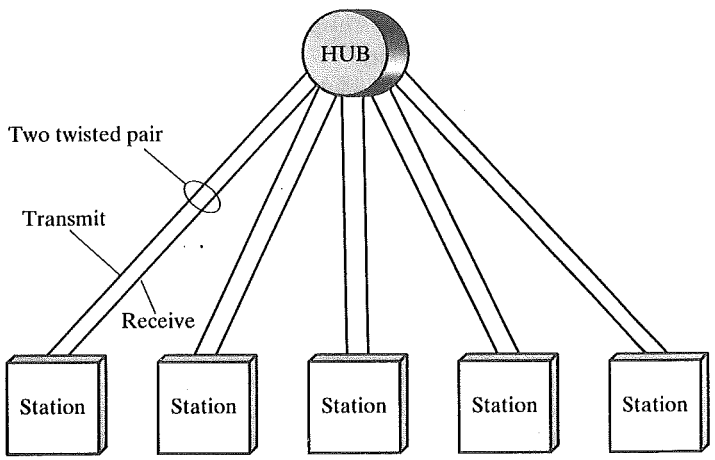


FIGURE 12.13 Twisted-pair star topology.

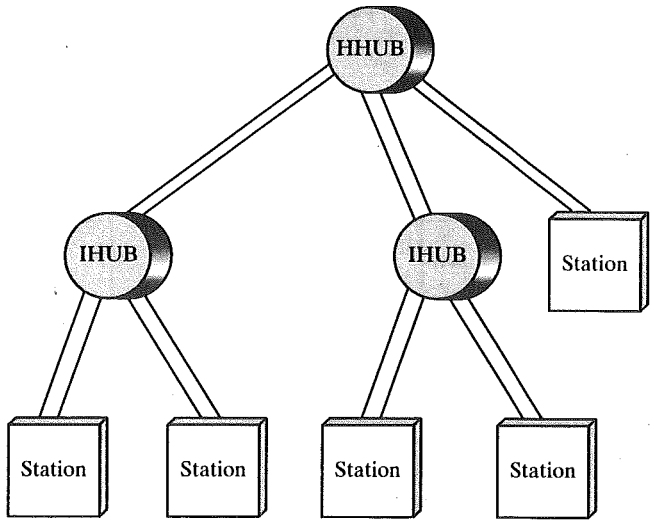


FIGURE 12.14 Two-level twisted-pair star topology.

optical fibers. Light that is input to one of the fibers on one side of the coupler is equally divided among, and output through, all the fibers on the other side. To form a network, each device is connected to the coupler with two fibers, one for transmit and one for receive (Figure 12.16). All of the transmit fibers enter the coupler on one side, and all of the receive fibers exit on the other side. Thus, although the arrangement is physically a star, it acts like a bus: A transmission from any one device is received by all other devices, and if two devices transmit at the same time, there will be a collision.

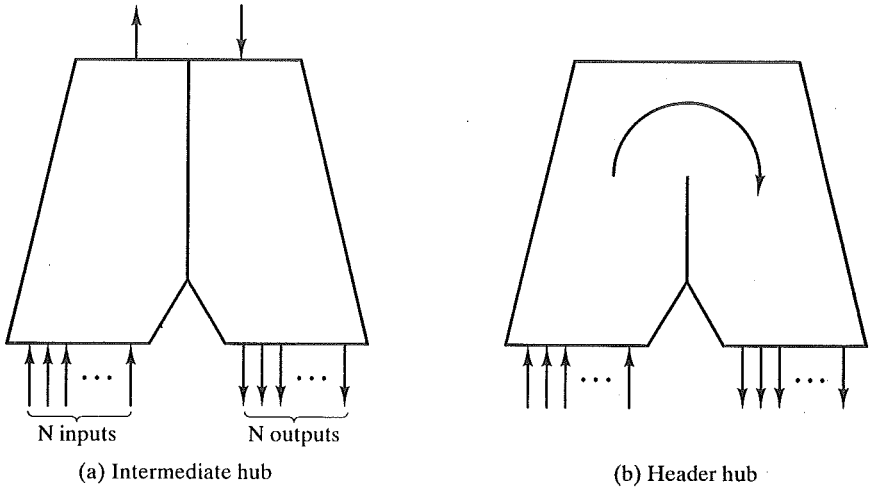


FIGURE 12.15 Intermediate and header hubs.

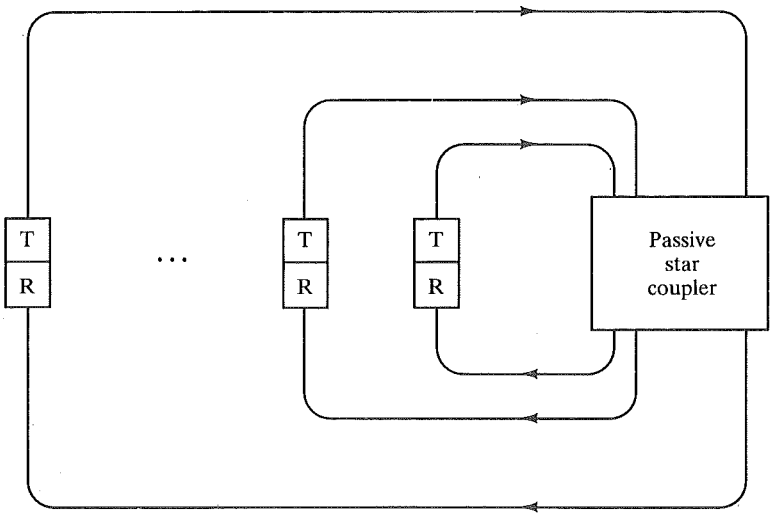


FIGURE 12.16 Optical fiber passive star configuration.

Two methods of fabrication of the star coupler have been pursued: the biconic fused coupler and the mixing rod coupler. In the biconic fused coupler, the fibers are bundled together and heated with an oxyhydrogen flame before being pulled into a biconical tapered shape. That is, the rods come together into a fused mass that tapers into a conical shape and then expands back out again. (The mixing rod approach begins in the same fashion.) Then, the biconical taper is cut at the waist and a cylindrical rod is inserted between the tapers and fused to the two cut ends. This latter technique allows the use of a less-narrow waist, and it is easier to fabricate.

Commercially available passive star couplers can support a few tens of stations at a radial distance of up to a kilometer or more. The limitations on number of stations and distances are imposed by the losses in the network. The attenuation that will occur in the network consists of the following components:

- **Optical connector losses.** Connectors are used to splice together cable segments for increased length. Typical connector losses are 1.0 to 1.5 dB per connector. A typical passive star network will have from 0 to 4 connectors in a path from transmitter to receiver, for a total maximum attenuation of 4 to 6 dB.
- **Optical cable attenuation.** Typical cable attenuation for the cable that has been used in these systems ranges from 3 to 6 dB per kilometer.
- **Optical power division in the coupler.** The coupler divides the optical power from one transmission path equally among all reception paths. Expressed in decibels, the loss seen by any node is 10 log N, where N is the number of nodes. For example, the effective loss in a 16-port coupler is about 12 dB.

12.5 WIRELESS LANs

In just the past few years, wireless LANs have come to occupy a significant niche in the local area network market. Increasingly, organizations are finding that wireless LANs are an indispensable adjunct to traditional wired LANs, as they satisfy requirements for mobility, relocation, ad hoc networking, and coverage of locations difficult to wire.

As the name suggests, a wireless LAN is one that makes use of a wireless transmission medium. Until relatively recently, wireless LANs were little used; the reasons for this included high prices, low data rates, occupational safety concerns, and licensing requirements. As these problems have been addressed, the popularity of wireless LANs has grown rapidly.

In this section, we first look at the requirements for and advantages of wireless LANs, and then preview the key approaches to wireless LAN implementation.

Wireless LANs Applications

[PAHL95a] lists four application areas for wireless LANs: LAN extension, cross-building interconnect, nomadic access, and ad hoc networks. Let us consider each of these in turn.

LAN Extension

Early wireless LAN products, introduced in the late 1980s, were marketed as substitutes for traditional wired LANs. A wireless LAN saves the cost of the installation of LAN cabling and eases the task of relocation and other modifications to network structure. However, this motivation for wireless LANs was overtaken by events. First, as awareness of the need for LAN became greater, architects designed new buildings to include extensive prewiring for data applications. Second, with advances in data transmission technology, there has been an increasing reliance on twisted pair cabling for LANs and, in particular, Category 3 unshielded twisted pair. Most older building are already wired with an abundance of Category 3 cable. Thus, the use of a wireless LAN to replace wired LANs has not happened to any great extent.

However, in a number of environments, there is a role for the wireless LAN as an alternative to a wired LAN. Examples include buildings with large open areas, such as manufacturing plants, stock exchange trading floors, and warehouses; historical buildings with insufficient twisted pair and in which drilling holes for new wiring is prohibited; and small offices where installation and maintenance of wired LANs is not economical. In all of these cases, a wireless LAN provides an effective and more attractive alternative. In most of these cases, an organization will also have a wired LAN to support servers and some stationary workstations. For example, a manufacturing facility typically has an office area that is separate from the factory floor but which must be linked to it for networking purposes. Therefore, typically, a wireless LAN will be linked into a wired LAN on the same premises. Thus, this application area is referred to as LAN extension.

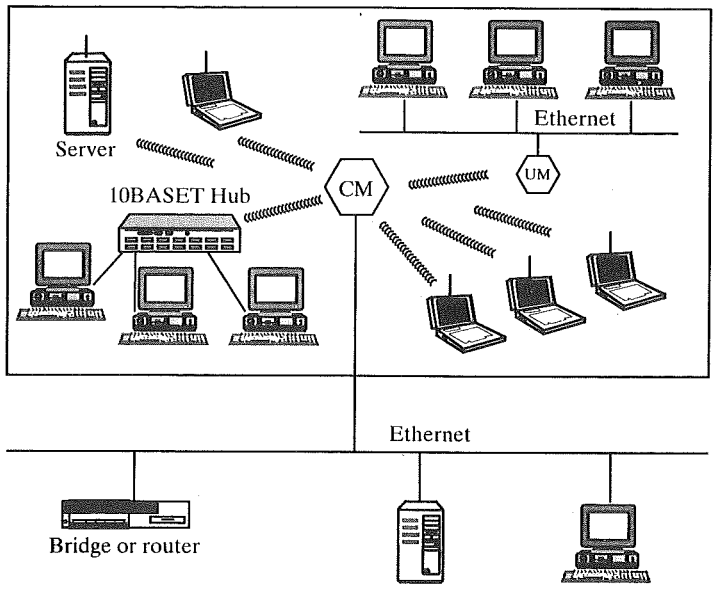


FIGURE 12.17 Example single-cell wireless LAN configuration.

Figure 12.17 indicates a simple wireless LAN configuration that is typical of many environments. There is a backbone wired LAN, such as Ethernet, that supports servers, workstations, and one or more bridges or routers to link with other networks. In addition there is a control module (CM) that acts as an interface to a wireless LAN. The control module includes either bridge or router functionality to link the wireless LAN to the backbone. In addition, it includes some sort of access control logic, such as a polling or token-passing scheme, to regulate the access from the end systems. Note that some of the end systems are standalone devices, such as a workstation or a server. In addition, hubs or other user modules (UM) that control a number of stations off a wired LAN may also be part of the wireless LAN configuration.

The configuration of Figure 12.17 can be referred to as a single-cell wireless LAN; all of the wireless end systems are within range of a single control module. Another common configuration, suggested by Figure 12.18, is a multiple-cell wireless LAN. In this case, there are multiple control modules interconnected by a wired LAN. Each control module supports a number of wireless end systems within its transmission range. For example, with an infrared LAN, transmission is limited to a single room; therefore, one cell is needed for each room in an office building that requires wireless support.

Cross-Building Interconnect

Another use of wireless LAN technology is to connect LANs in nearby buildings, be they wired or wireless LANs. In this case, a point-to-point wireless link is used between two buildings. The devices so connected are typically bridges or routers. This single point-to-point link is not a LAN per se, but it is usual to include this application under the heading of wireless LAN.

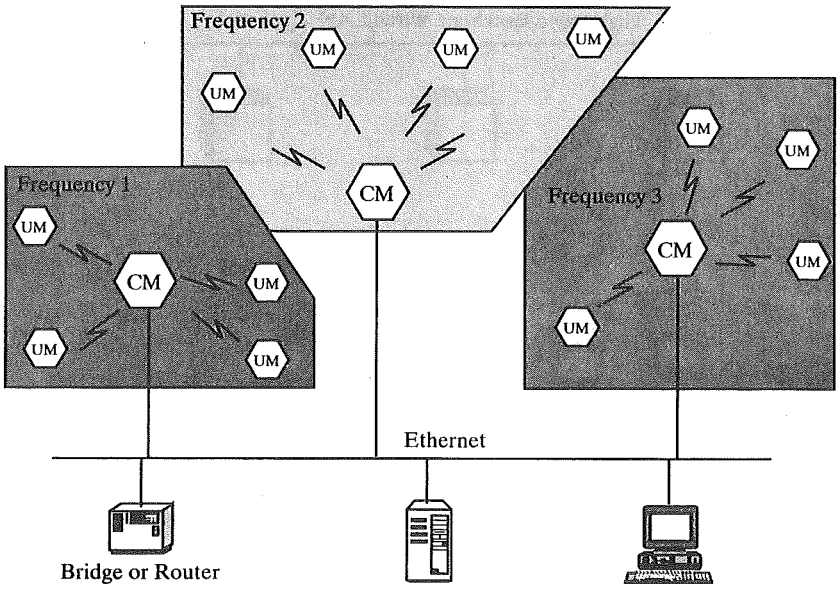


FIGURE 12.18 Example multiple-cell wireless LAN configuration.

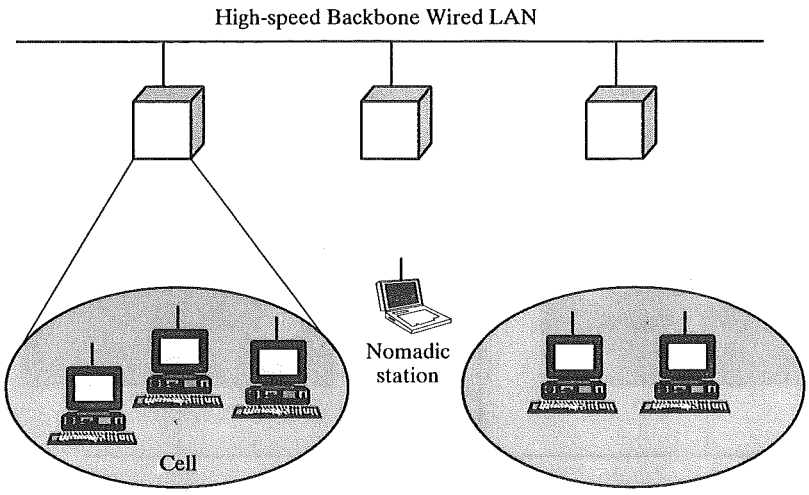
Nomadic Access

Nomadic access provides a wireless link between a LAN hub and a mobile data terminal equipped with an antenna, such as a laptop computer or notepad computer. One example of the utility of such a connection is to enable an employee returning from a trip to transfer data from a personal portable computer to a server in the office. Nomadic access is also useful in an extended environment such as a campus or a business operating out of a cluster of buildings. In both of these cases, users may move around with their portable computers and may wish access to the servers on a wired LAN from various locations.

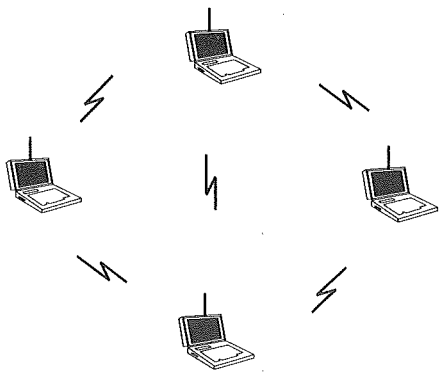
Ad Hoc Networking

An ad hoc network is a peer-to-peer network (no centralized server) set up temporarily to meet some immediate need. For example, a group of employees, each with a laptop or palmtop computer, may convene in a conference room for a business or classroom meeting. The employees link their computers in a temporary network just for the duration of the meeting.

Figure 12.19 suggests the differences between an ad hoc wireless LAN and a wireless LAN that supports LAN extension and nomadic access requirements. In the former case, the wireless LAN forms a stationary infrastructure consisting of one or more cells with a control module for each cell. Within a cell, there may be a number of stationary end systems. Nomadic stations can move from one cell to another. In contrast, there is no infrastructure for an ad hoc network. Rather, a peer collection of stations within range of each other may dynamically configure themselves into a temporary network.



(a) Infrastructure wireless LAN



(b) Ad hoc LAN

FIGURE 12.19 Wireless LAN configurations.

Wireless LAN Requirements

A wireless LAN must meet the same sort of requirements typical of any LAN, including high capacity, ability to cover short distances, full connectivity among attached stations, and broadcast capability. In addition, there are a number of requirements specific to the wireless LAN environment. The following are among the most important requirements for wireless LANs:

- **Throughput.** The medium access control protocol should make as efficient use as possible of the wireless medium to maximize capacity.
- **Number of nodes.** Wireless LANs may need to support hundreds of nodes across multiple cells.
- **Connection to backbone LAN.** In most cases, interconnection with stations on a wired backbone LAN is required. For infrastructure wireless LANs, this is easily accomplished through the use of control modules that connect to both

types of LANs. There may also need to be accommodation for mobile users and ad hoc wireless networks.

- **Service area.** A typical coverage area for a wireless LAN may be up to a 300 to 1000 foot diameter.
- **Battery power consumption.** Mobile workers use battery-powered workstations that need to have a long battery life when used with wireless adapters. This suggests that a MAC protocol that requires mobile nodes to constantly monitor access points or to engage in frequent handshakes with a base station is inappropriate.
- **Transmission robustness and security.** Unless properly designed, a wireless LAN may be interference-prone and easily eavesdropped upon. The design of a wireless LAN must permit reliable transmission even in a noisy environment and should provide some level of security from eavesdropping.
- **Collocated network operation.** As wireless LANs become more popular, it is quite likely for two of them to operate in the same area or in some area where interference between the LANs is possible. Such interference may thwart the normal operation of a MAC algorithm and may allow unauthorized access to a particular LAN.
- **License-free operation.** Users would prefer to buy and operate wireless LAN products without having to secure a license for the frequency band used by the LAN.
- **Handoff/roaming.** The MAC protocol used in the wireless LAN should enable mobile stations to move from one cell to another.
- **Dynamic configuration.** The MAC addressing and network management aspects of the LAN should permit dynamic and automated addition, deletion, and relocation of end systems without disruption to other users.

Wireless LAN Technology

Wireless LANs are generally categorized according to the transmission technique that is used. All current wireless LAN products fall into one of the following categories:

- **Infrared (IR) LANs.** An individual cell of an IR LAN is limited to a single room, as infrared light does not penetrate opaque walls.
- **Spread Spectrum LANs.** This type of LAN makes use of spread spectrum transmission technology. In most cases, these LANs operate in the ISM (Industrial, Scientific, and Medical) bands, so that no FCC licensing is required for their use in the U.S.
- **Narrowband Microwave.** These LANs operate at microwave frequencies but do not use spread spectrum. Some of these products operate at frequencies that require FCC licensing, while others use one of the unlicensed ISM bands.

Table 12.6 summarizes some of the key characteristics of these three technologies.

TABLE 12.6 Comparison of wireless LAN technologies.

	Infrared		Spread Spectrum		Radio
	Diffused Infrared	Directed Beam Infrared	Frequency Hopping	Direct Sequence	Narrowband Microwave
Data rate (Mbps)	1–4	1–0	1–3	2–20	5–10
Mobility	Stationary/mobile	Stationary with LOS	Mobile	Stationary/mobile	
Range (ft)	50–200	80	100–300	100–800	40–130
Detectability	Negligible		Little		Some
Wavelength/ frequency	λ: 800 – 900 nm		ISM bands: 902 – 928 MHz 2.4 – 2.4835 GHz 5.725 – 5.85 GHz		18.825 – 19.205 GHz or ISM band
Modulation technique	OOK		GFSK	QPSK	FS/QPSK
Radiated power	N/A		<1W		25 mW
Access method	CSMA	Token Ring, CSMA	CSMA		Reservation ALOHA, CSMA
License required	No		No		Yes unless ISM

12.6 RECOMMENDED READING

The literature on LANs and MANs is vast. The material in this chapter is covered in much more depth in [STAL97].

[MART94] and [MADR94] are book-length treatments of LANs. [SADI95] and [KESS92] cover MANs.

[PAHL95a] and [BANT94] are excellent survey articles on wireless LANs. Two book-length treatments are noteworthy: [SANT94] focuses on the technology of wireless LAN components and on signal encoding techniques; [DAVI95] addresses applications for wireless LANs as well as configuration and management issues.

BANT94 Bantz, D. and Bauchot, F. "Wireless LAN Design Alternatives." *IEEE Network*, March/April, 1994.

DAVI95 Davis, P. and McGuffin, C. *Wireless Local Area Networks*. New York: McGraw-Hill, 1995.

KESS92 Kessler, G. and Train, D. *Metropolitan Area Networks: Concepts, Standards, and Services*. New York: McGraw-Hill, 1992.

MADR94 Madron, T. *Local Area Networks: New Technologies, Emerging Standards*. New York: Wiley, 1994.

MART94 Martin, J., Chapman, K., and Leben, J. *Local Area Networks: Architectures and Implementations*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

PAHL95a Pahlavan, K., Probert, T., and Chase, M. "Trends in Local Wireless Networks." *IEEE Communications Magazine*, March 1995.

SADI95 Sadiku, M. *Metropolitan Area Networks*. Boca Raton, FL: CRC Press, 1995.

SANT94 Santamaria, A. and Lopez-Hernandez, F. (*editors*). *Wireless LAN Systems*. Boston MA: Artech House, 1994.

STAL97 Stallings, W. *Local and Metropolitan Area Networks, Fifth Edition*. Upper Saddle River, NJ: Prentice Hall, 1997.

Recommended Web Site

- http://web.syr.edu/~jmwobus/lans: This site has links to most important sources of LAN information on the Internet, including all of the related FAQs.

12.7 PROBLEMS

12.1 Could HDLC be used as a data link control protocol for a LAN? If not, what is missing?

12.2 An asynchronous device, such as a teletype, transmits characters one at a time with unpredictable delays between characters. What problems, if any, do you foresee if such a device is connected to a local network and allowed to transmit at will (subject to gaining access to the medium)? How might such problems be resolved?

12.3 Consider the transfer of a file containing one million characters from one station to another. What is the total elapsed time and effective throughput for the following cases:

a. A circuit-switched, star topology local network. Call setup time is negligible, and the data rate on the medium is 64 kbps.

b. A bus topology local network with two stations a distance D apart, a data rate of B bps, and a packet size P with 80 bits of overhead. Each packet is acknowledged with an 88-bit packet before the next is sent. The propagation speed on the bus is 200 m/μs. Solve for

(1) $D = 1$ km, $B = 1$ Mbps, $P = 256$ bits
(2) $D = 1$ km, $B = 10$ Mbps, $P = 256$ bits
(3) $D = 10$ km, $B = 1$ Mbps, $P = 256$ bits
(4) $D = 1$ km, $B = 50$ Mbps, $P = 10{,}000$ bits

c. A ring topology with a total circular length of $2D$, with the two stations a distance D apart. Acknowledgment is achieved by allowing a packet to circulate past the destination station, back to the source station. There are N repeaters on the ring, each of which introduces a delay of one bit time. Repeat the calculation for each of b(1) through b(4) for $N = 10; 100; 1000$.

12.4 Consider a baseband bus with a number of equally spaced stations with a data rate of 10 Mbps and a bus length of 1 km.
a. What is the average time to send a frame of 1000 bits to another station, measured from the beginning of transmission to the end of reception? Assume a propagation speed of 200 m/μs.
b. If two stations begin to transmit at exactly the same time, their packets will interfere with each other. If each transmitting station monitors the bus during transmission, how long before it notices an interference, in seconds? In bit times?

12.5 Repeat Problem 12.4 for a data rate of 1 Mbps.

12.6 Repeat Problems 12.4 and 12.5 for
a. Broadband bus
b. Broadband tree consisting of 10 cables each of length 100 m emanating from a headend.

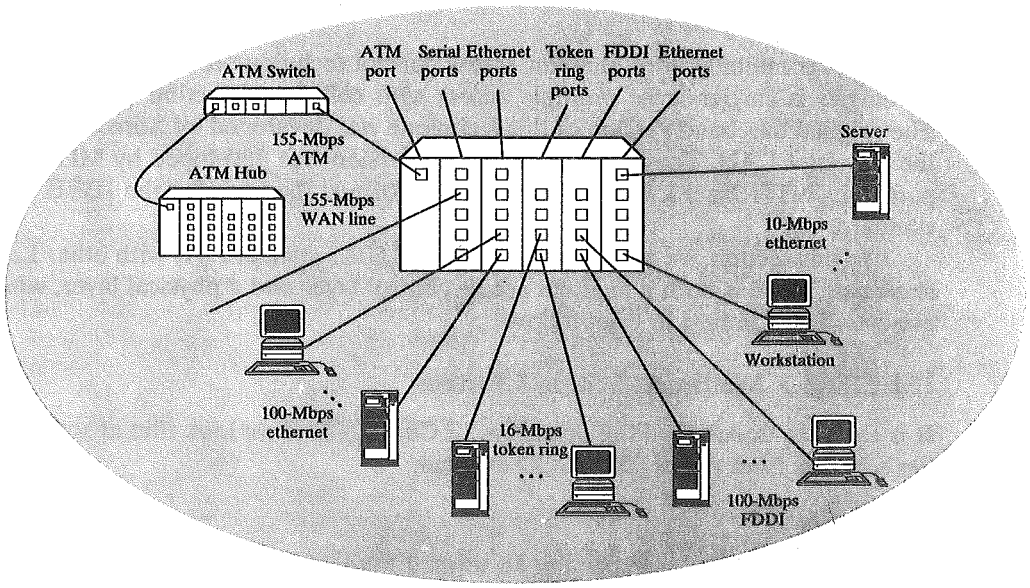
12.7 At a propagation speed of 200 m/μs, what is the effective length added to a ring by a bit delay at each repeater:
a. At 1 Mbps?
b. At 100 Mbps?

12.8 A tree-topology local network is to be provided that spans two buildings. If permission can be obtained to string cable between the two buildings, one continuous tree layout will be used. Otherwise, each building will have an independent tree-topology network, and a point-to-point link will connect a special communications station on one network with a communications station on the other network. What functions must the communications stations perform? Repeat for ring and star.

12.9 System A consists of a single ring with 300 stations, one per repeater. System B consists of three 100-station rings linked by a bridge. If the probability of a link failure is P_1, a repeater failure is P_r, and a bridge failure is P_b, derive an expression for parts (a) through (d):
a. Probability of failure of system A.
b. Probability of complete failure of system B.
c. Probability that a particular station will find the network unavailable, for systems A and B.
d. Probability that any two stations, selected at random, will be unable to communicate, for systems A and B
e. Compute values for parts (a) through (d) for $P_1 = P_b = P_r = 10^{-2}$.

CHAPTER 13

LAN SYSTEMS



We now move to a consideration of specific LAN systems. As was mentioned in Chapter 12, the medium access control technique and topology are key characteristics used in the classification of LANs and in the development of standards. The following systems are discussed in this chapter:[1]

- Ethernet and Fast Ethernet (CSMA/CD)
- Token Ring/FDDI
- 100VG-AnyLAN
- ATM LANs
- Fibre Channel
- Wireless LANs

13.1 ETHERNET AND FAST ETHERNET (CSMA/CD)

The most commonly used medium access control technique for bus/tree and star topologies is carrier-sense multiple access with collision detection (CSMA/CD). The original baseband version of this technique was developed by Xerox as part of the Ethernet LAN. The original broadband version was developed by MITRE as part of its MITREnet LAN. All of this work formed the basis for the IEEE 802.3 standard.

In this section, we will focus on the IEEE 802.3 standard. As with other LAN standards, there is both a medium access control layer and a physical layer, which are considered in turn in what follows.

IEEE 802.3 Medium Access Control

It is easier to understand the operation of CSMA/CD if we look first at some earlier schemes from which CSMA/CD evolved.

Precursors

CSMA/CD and its precursors can be termed random access, or contention, techniques. They are random access in the sense that there is no predictable or scheduled time for any station to transmit; station transmissions are ordered randomly. They exhibit contention in the sense that stations contend for time on the medium.

The earliest of these techniques, known as ALOHA, was developed for packet radio networks. However, it is applicable to any shared transmission medium. ALOHA, or pure ALOHA as it is sometimes called, is a true free-for-all. Whenever a station has a frame to send, it does so. The station then listens for an amount of time equal to the maximum possible round-trip propagation delay on the network (twice the time it takes to send a frame between the two most widely separated stations) plus a small fixed time increment. If the station hears an acknowl-

[1] Two other systems illustrated in Figure 12.2, DQDB MANs and Token Bus, are not discussed in this chapter due to space constraints. These systems are not as widely used as the others covered in this chapter.

edgment during that time, fine; otherwise, it resends the frame. If the station fails to receive an acknowledgment after repeated transmissions, it gives up. A receiving station determines the correctness of an incoming frame by examining a frame-check-sequence field, as in HDLC. If the frame is valid and if the destination address in the frame header matches the receiver's address, the station immediately sends an acknowledgment. The frame may be invalid due to noise on the channel or because another station transmitted a frame at about the same time. In the latter case, the two frames may interfere with each other at the receiver so that neither gets through; this is known as a *collision*. If a received frame is determined to be invalid, the receiving station simply ignores the frame.

ALOHA is as simple as can be, and pays a penalty for it. Because the number of collisions rise rapidly with increased load, the maximum utilization of the channel is only about 18% (see [STAL97]).

To improve efficiency, a modification of ALOHA, known as slotted ALOHA, was developed. In this scheme, time on the channel is organized into uniform slots whose size equals the frame transmission time. Some central clock or other technique is needed to synchronize all stations. Transmission is permitted to begin only at a slot boundary. Thus, frames that do overlap will do so totally. This increases the maximum utilization of the system to about 37%.

Both ALOHA and slotted ALOHA exhibit poor utilization. Both fail to take advantage of one of the key properties of both packet radio and LANs, which is that propagation delay between stations is usually very small compared to frame transmission time. Consider the following observations. If the station-to-station propagation time is large compared to the frame transmission time, then, after a station launches a frame, it will be a long time before other stations know about it. During that time, one of the other stations may transmit a frame; the two frames may interfere with each other and neither gets through. Indeed, if the distances are great enough, many stations may begin transmitting, one after the other, and none of their frames get through unscathed. Suppose, however, that the propagation time is small compared to frame transmission time. In that case, when a station launches a frame, all the other stations know it almost immediately. So, if they had any sense, they would not try transmitting until the first station was done. Collisions would be rare because they would occur only when two stations began to transmit almost simultaneously. Another way to look at it is that a short delay time provides the stations with better feedback about the state of the network; this information can be used to improve efficiency.

The foregoing observations led to the development of carrier-sense multiple access (CSMA). With CSMA, a station wishing to transmit first listens to the medium to determine if another transmission is in progress (carrier sense). If the medium is in use, the station must wait. If the medium is idle, the station may transmit. It may happen that two or more stations attempt to transmit at about the same time. If this happens, there will be a collision; the data from both transmissions will be garbled and not received successfully. To account for this, a station waits a reasonable amount of time, after transmitting, for an acknowledgment, taking into account the maximum round-trip propagation delay, and the fact that the acknowledging station must also contend for the channel in order to respond. If there is no acknowledgment, the station assumes that a collision has occurred and retransmits.

One can see how this strategy would be effective for networks in which the average frame transmission time is much longer than the propagation time. Collisions can occur only when more than one user begins transmitting within a short time (the period of the propagation delay). If a station begins to transmit a frame, and there are no collisions during the time it takes for the leading edge of the packet to propagate to the farthest station, then there will be no collision for this frame because all other stations are now aware of the transmission.

The maximum utilization achievable using CSMA can far exceed that of ALOHA or slotted ALOHA. The maximum utilization depends on the length of the frame and on the propagation time; the longer the frames or the shorter the propagation time, the higher the utilization. This subject is explored in Appendix 13A.

With CSMA, an algorithm is needed to specify what a station should do if the medium is found busy. The most common approach, and the one used in IEEE 802.3, is the *1-persistent technique.* A station wishing to transmit listens to the medium and obeys the following rules:

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, continue to listen until the channel is sensed idle; then transmit immediately.

If two or more stations are waiting to transmit, a collision is guaranteed. Things get sorted out only after the collision.

Description of CSMA/CD

CSMA, although more efficient than ALOHA or slotted ALOHA, still has one glaring inefficiency: When two frames collide, the medium remains unusable for the duration of transmission of both damaged frames. For long frames, compared to propagation time, the amount of wasted capacity can be considerable. This waste can be reduced if a station continues to listen to the medium while transmitting. This leads to the following rules for CSMA/CD:

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, continue to listen until the channel is idle, then transmit immediately.
3. If a collision is detected during transmission, transmit a brief jamming signal to assure that all stations know that there has been a collision and then cease transmission.
4. After transmitting the jamming signal, wait a random amount of time, then attempt to transmit again. (Repeat from step 1.)

Figure 13.1 illustrates the technique for a baseband bus. At time t_0, station A begins transmitting a packet addressed to D. At t_1, both B and C are ready to transmit. B senses a transmission and so defers. C, however, is still unaware of A's transmission and begins its own transmission. When A's transmission reaches C, at t_2, C detects the collision and ceases transmission. The effect of the collision propagates

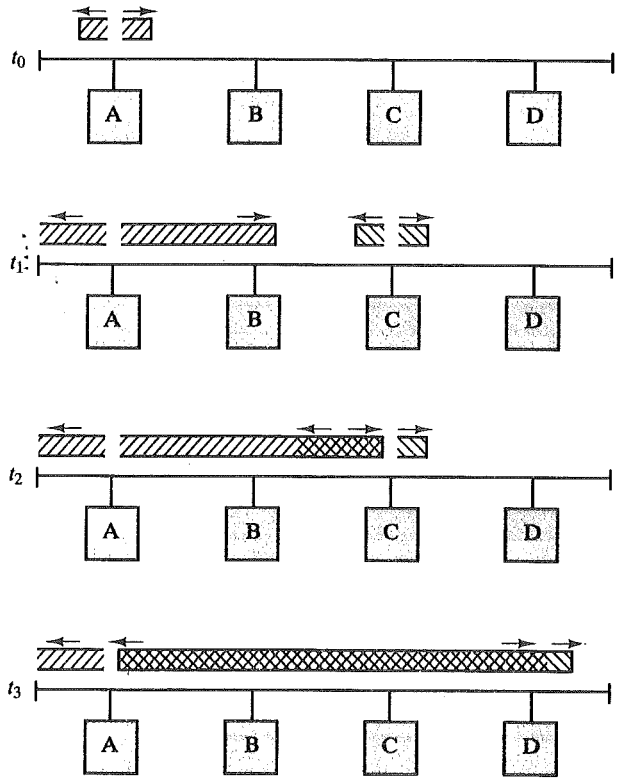


FIGURE 13.1 CSMA/CD operation.

back to A, where it is detected some time later, t_3, at which time A ceases transmission.

With CSMA/CD, the amount of wasted capacity is reduced to the time it takes to detect a collision. Question: how long does that take? Let us consider first the case of a baseband bus and consider two stations as far apart as possible. For example, in Figure 13.1, suppose that station A begins a transmission and that just before that transmission reaches D, D is ready to transmit. Because D is not yet aware of A's transmission, it begins to transmit. A collision occurs almost immediately and is recognized by D. However, the collision must propagate all the way back to A before A is aware of the collision. By this line of reasoning, we conclude that the amount of time that it takes to detect a collision is no greater than twice the end-to-end propagation delay. For a broadband bus, the delay is even longer. Figure 13.2 shows a dual-cable system. This time, the worst case occurs for two stations as close together as possible and as far as possible from the headend. In this case, the maximum time to detect a collision is four times the propagation delay from an end of the cable to the headend.

An important rule followed in most CSMA/CD systems, including the IEEE standard, is that frames should be long enough to allow collision detection prior to the end of transmission. If shorter frames are used, then collision detection does not

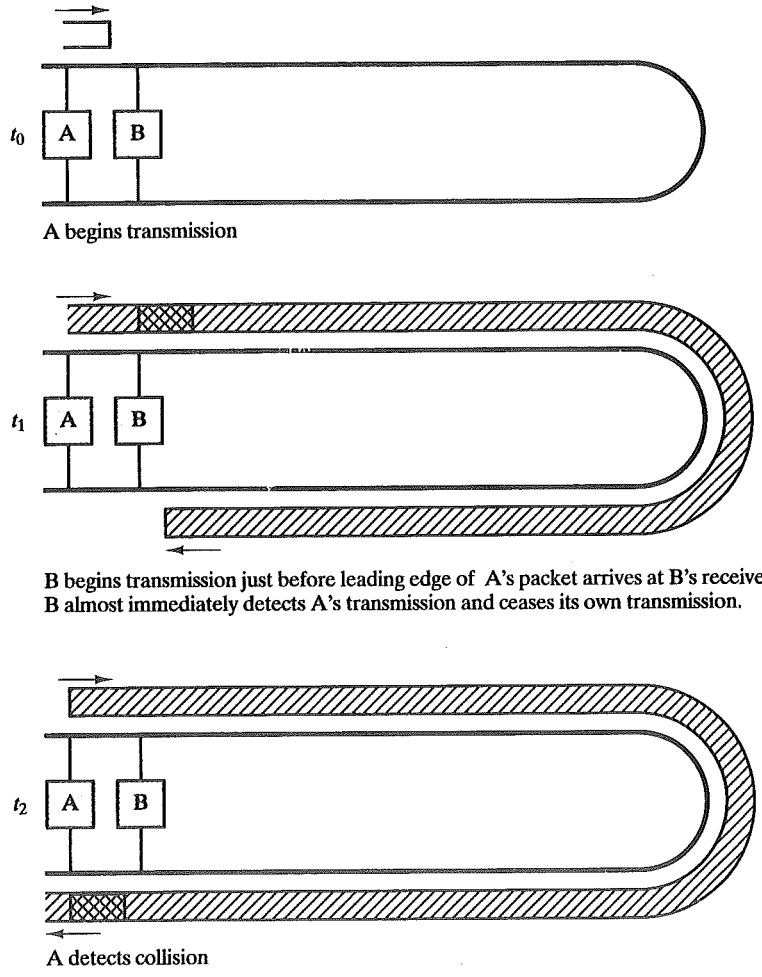


FIGURE 13.2 Broadband collision detection timing.

occur, and CSMA/CD exhibits the same performance as the less efficient CSMA protocol.

Although the implementation of CSMA/CD is substantially the same for baseband and broadband, there are differences. One is the means for performing carrier sense; for baseband systems, this is done by detecting a voltage pulse train. For broadband, the RF carrier is detected.

Collision detection also differs for the two systems. For baseband, a collision should produce substantially higher voltage swings than those produced by a single transmitter. Accordingly, the IEEE standard dictates that the transmitter will detect a collision if the signal on the cable at the transmitter tap point exceeds the maximum that could be produced by the transmitter alone. Because a transmitted signal attenuates as it propagates, there is a potential problem: If two stations far apart are transmitting, each station will receive a greatly attenuated signal from the other. The signal strength could be so small that when it is added to the transmitted

signal at the transmitter tap point, the combined signal does not exceed the CD threshold. For this reason, among others, the IEEE standard restricts the maximum length of coaxial cable to 500 m for 10BASE5 and to 200 m for 10BASE2.

A much simpler collision detection scheme is possible with the twisted pair star-topology approach (Figure 13.1). In this case, collision detection is based on logic rather than on sensing voltage magnitudes. For any hub, if there is activity (signal) on more than one input, a collision is assumed. A special signal called the collision presence signal is generated. This signal is generated and sent out as long as activity is sensed on any of the input lines. This signal is interpreted by every node as an occurrence of a collision.

There are several possible approaches to collision detection in broadband systems. The most common of these is to perform a bit-by-bit comparison between transmitted and received data. When a station transmits on the inbound channel, it begins to receive its own transmission on the outbound channel after a propagation delay to the headend and back. Note the similarity to a satellite link. Another approach, for split systems, is for the headend to perform detection based on garbled data.

MAC Frame

Figure 13.3 depicts the frame format for the 802.3 protocol; it consists of the following fields:

- **Preamble.** A 7-octet pattern of alternating 0s and 1s used by the receiver to establish bit synchronization.
- **Start frame delimiter.** The sequence 10101011, which indicates the actual start of the frame and which enables the receiver to locate the first bit of the rest of the frame.
- **Destination address (DA).** Specifies the station(s) for which the frame is intended. It may be a unique physical address, a group address, or a global address. The choice of a 16- or 48-bit address length is an implementation decision, and must be the same for all stations on a particular LAN.
- **Source address (SA).** Specifies the station that sent the frame.
- **Length.** Length of the LLC data field.
- **LLC data.** Data unit supplied by LLC.
- **Pad.** Octets added to ensure that the frame is long enough for proper CD operation.

Octets	7	1	2 or 6	2 or 6	2	≥0	≥0	4
	Preamble	SFD	DA	SA	Length	LLC data	Pad	FCS

LEGEND

SFD = Start-frame delimiter
DA = Destination address
SA = Source address
FCS = Frame-check sequence

FIGURE 13.3 IEEE 802.3 frame format.

- **Frame check sequence (FCS).** A 32-bit cyclic redundancy check, based on all fields except the preamble, the SFD, and the FCS.

IEEE 802.3 10-Mbps Specifications (Ethernet)

The IEEE 802.3 committee has been the most active in defining alternative physical configurations; this is both good and bad. On the good side, the standard has been responsive to evolving technology. On the bad side, the customer, not to mention the potential vendor, is faced with a bewildering array of options. However, the committee has been at pains to ensure that the various options can be easily integrated into a configuration that satisfies a variety of needs. Thus, the user that has a complex set of requirements may find the flexibility and variety of the 802.3 standard to be an asset.

To distinguish among the various implementations that are available, the committee has developed a concise notation:

<data rate in Mbps.<signaling method><maximum segment length in hundreds of meters>

The defined alternatives are:[2]

- 10BASE5
- 10BASE2
- 10BASE-T
- 10BROAD36
- 10BASE-F

Note that 10BASE-T and 10-BASE-F do not quite follow the notation; "T" stands for twisted pair, and "F" stands for optical fiber. Table 13.1 summarizes these options. All of the alternatives listed in the table specify a data rate of 10 Mbps. In addition to these alternatives, there are several versions that operate at 100 Mbps; these are covered later in this section.

10BASE5 Medium Specification

10BASE5 is the original 802.3 medium specification and is based on directly on Ethernet. 10BASE5 specifies the use of 50-ohm coaxial cable and uses Manchester digital signaling.[3] The maximum length of a cable segment is set at 500 meters. The length of the network can be extended by the use of repeaters, which are transparent to the MAC level; as they do no buffering, they do not isolate one segment from another. So, for example, if two stations on different segments attempt to transmit at the same time, their transmissions will collide. To avoid looping, only one path of segments and repeaters is allowed between any two stations. The standard allows a

[2] There is also a 1BASE-T alternative that defines a 1-Mbps twisted pair system using a star topology; this is considered obsolete although it is contained in the most recent version of the standard.

[3] See Section 4.1.