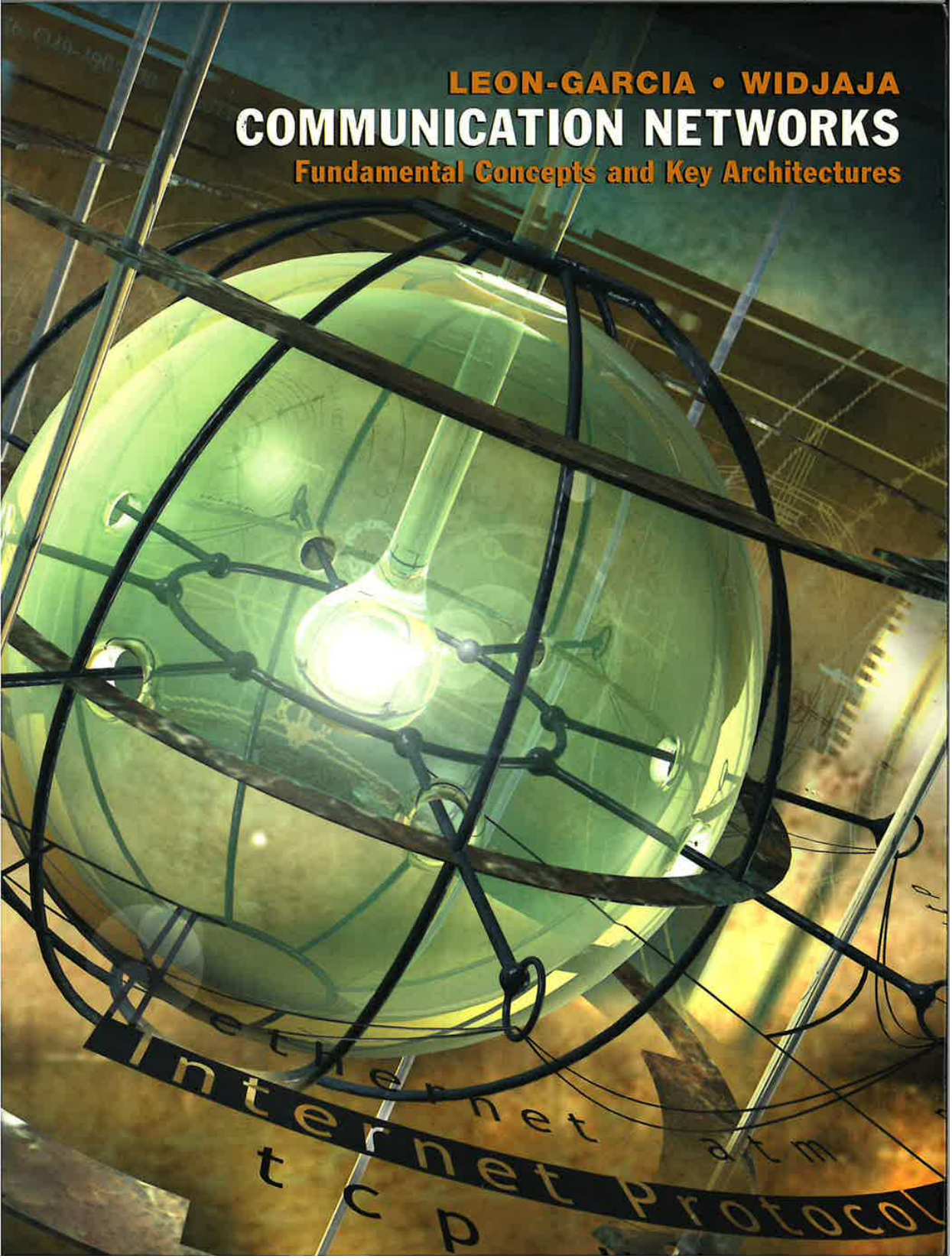
LEON-GARCIA • WIDJAJA

COMMUNICATION NETWORKS

Fundamental Concepts and Key Architectures



Nokia of America Corporation
IPR2023-00680, Ex. 1015
Ex. 1015.1

Communication Networks

Ex. 1015.2

McGraw-Hill Series in Computer Science

SENIOR CONSULTING EDITOR

C.L. Liu, University of Illinois at Urbana-Champaign

CONSULTING EDITOR

Allen B. Tucker, Bowdoin College

Fundamentals of Computing and Programming
Computer Organization and Architecture
Computers in Society/Ethics
Systems and Languages
Theoretical Foundations
Software Engineering and Database
Artificial Intelligence
Networks, Parallel and Distributed Computing
Graphics and Visualization
The MIT Electrical and Computer Science Series

McGraw-Hill Series in Electrical and Computer Engineering

SENIOR CONSULTING EDITOR

Stephen W. Director, University of Michigan, Ann Arbor

Circuits and Systems
Communications and Signal Processing
Computer Engineering
Control Theory and Robotics
Electromagnetics
Electronics and VLSI Circuits
Introductory
Power
Antennas, Microwaves, and Radar

PREVIOUS CONSULTING EDITORS

Ronald N. Bracewell, Colin Cherry, James F. Gibbons, Willis W. Harman, Hubert Heffner, Edward W. Herold, John G. Linvill, Simon Ramo, Ronald A. Rohrer, Anthony E. Siegman, Charles Susskind, Frederick E. Terman, John G. Truxal, Ernst Weber, and John R. Whinnery

Communication Networks

Fundamental Concepts and Key Architectures

Alberto Leon-Garcia

University of Toronto

Indra Widjaja

Boston Burr Ridge, IL Dubuque, IA Madison, WI
New York San Francisco St. Louis
Bangkok Bogotá Caracas Lisbon London Madrid Mexico City
Milan New Delhi Seoul Singapore Sydney Taipei Toronto

Ex. 1015.4

McGraw-Hill Higher Education
A Division of The McGraw-Hill Companies

COMMUNICATION NETWORKS
Fundamental Concepts and Key Architectures

Copyright © 2000 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 0 9 8 7 6 5 4 3 2 1 0 9

ISBN 0-07-022839-6

Vice president/Editor-in-chief: *Kevin T. Kane*
Publisher: *Thomas Casson*
Executive editor: *Elizabeth A. Jones*
Senior developmental editor: *Kelley Butcher*
Senior marketing manager: *John T. Wannemacher*
Project manager: *Jim Labeots*
Senior production supervisor: *Heather D. Burbridge*
Freelance design coordinator: *Gino Cieslik*
Cover illustration: *Henk Dawson*
Supplement coordinator: *Rose M. Range*
Compositor: *Keyword Publishing Services*
Typeface: *10/12 Times Roman*
Printer: *R. R. Donnelley & Sons Company*

Library of Congress Cataloging-in-Publication Data
Leon-Garcia, Alberto.
Communication networks / Alberto Leon-Garcia, Indra Widjaja.
p. cm.
Includes bibliographical references and index.
ISBN 0-07-022839-6 (alk. paper)
1. Telecommunication systems. 2. Computer networks. I. Widjaja, Indra, 1960- II. Title.
TK5101.L46 2000
621.382′1–dc21

99-052271

http://www.mhhe.com

To my teachers
who taught me how to learn for the sake of learning,
and to my students
who taught me how to learn for the sake of teaching.
—alg

To my parents: with gratitude for your love and support.
—iw

Ex. 1015.6

PREFACE

OBJECTIVE

Communication networks have entered an era of fundamental change where market and regulatory forces have finally caught up with the relentless advance of technology, as evidenced by the following:

- The explosive growth of multimedia personal computing and the World Wide Web, demonstrating the value of network-based services.
- The deregulation of the telecommunications industry opening the door to new access network technologies (digital cellular systems, cable modems, high-speed DSL modems, direct broadcast satellite systems, satellite constellation networks, broadband wireless cable) that will cause telecommunications infrastructure to migrate towards a flexible packet-based backbone network technology.
- The explosion in available bandwidth due to optical transmission technology and the entry of new national and global backbone service providers.
- The emergence of the Internet suite of protocols as the primary means for providing ubiquitous connectivity across the emerging network of networks.
- The predominance of data traffic over voice traffic dictating that future networks will be designed for data, and that telephone voice service must eventually operate—possibly solely—over the Internet.

Thus, the main architectural elements of the network of networks that will emerge in the next ten years are becoming more evident. The purpose of this book is to introduce electrical engineering, computer engineering, and computer science students to fundamental network architecture concepts and to their application in these emerging networks.

TARGET COURSES

The book is designed for introductory one-semester or one-year courses in communication networks in the upper-level undergraduate and first-year graduate programs. The second half of the book can be used in more advanced courses that deal with the details of current network architectures. The book can also be used by engineering and computer professionals seeking an introduction to networking.

As prerequisites the book assumes a general knowledge of computer systems and programming, and elementary calculus. In certain parts of the text, knowledge of elementary probability is useful but not essential.

Ex. 1015.7

APPROACH AND CONTENT

Networks are extremely complex systems consisting of many components whose operation depends on many processes. To understand networks it is essential that students be exposed to *the big picture of networks* that allows them to see how the various parts of the network fit into one whole. We have designed the book so that students are presented with this big picture at the beginning of the book. The students then have a context in which to place the various topics as they progress through the book.

The book attempts to provide a *balanced view of all important elements of networking*. This is a very big challenge in the typical one-semester introductory course which has very limited time available. We have organized the book so that all the relevant topics can be covered at some minimum essential level of detail. Additional material is provided that allows the instructor to cover certain topics in greater depth.

The book is organized into four sections: the first section provides the big picture; the second section develops fundamental concepts; the third section deals with advanced topics and detailed network architectures; and in the fourth section two appendices provide important supporting material.

Big Picture First: Networks, Services, and Layered Architectures

This section begins in Chapter 1 with a discussion of network-based applications that the student is familiar with (World Wide Web, e-mail, telephone call, and home video entertainment). These examples are used to emphasize that modern networks must be designed to support a wide range of applications. We then discuss the evolution of telegraph, telephone, and computer networks, up to the present Internet. This historical discussion is used to identify the essential functions that are common to all networks. We show how there is usually more than one way to carry out a function, for example, connectionless versus circuit-switched transfer of information, and that the specific structure of a network is determined by a combination of technological, market, and regulatory factors at a given point in time.

The view of the network as a provider of services to applications is developed in Chapter 2. We consider the e-mail and Web browsing applications, and we explain the application layer protocols that support these, namely HTTP, SMTP, and DNS. We also explain how these protocols in turn make use of the communication services provided by TCP and UDP. Together these examples motivate the notion of layering, leading naturally to a discussion of the OSI reference model. A detailed example is used to show how Ethernet, PPP, IP, TCP, and UDP work together to support the application layer protocols. The key notions of addressing and encapsulation are developed in this example. Chapter 2 concludes with two optional sections: an introduction to sockets and an introduction to additional application layer protocols and to several TCP/IP utilities. We believe that the student will be familiar with some of the application layer topics,

Ex. 1015.8

and so Chapter 2 can serve as a bridge to the less visible topics relating to the internal operation of a network. Sockets and TCP/IP utilities provide the basis for very useful and practical exercises and experiments that provide students with some "hands on" networking experience.

Fundamental Network Architecture Concepts

The second section develops the fundamental concepts of network architecture, proceeding from the physical layer to the network layer. We complement the discussion of fundamental concepts with sections that explore trends in network architecture.

Chapter 3 deals with digital transmission including error detetection. We identify the bit rate requirements that applications impose on the network, and then we examine the transmission capabilities of existing and emerging networks. We introduce the relationship between bandwidth, bit rate, and signal-to-noise ratio, and then develop the basic digital transmission techniques, using modem standards as examples. The properties of various media (copper wires, coaxial cable, radio, optical fiber) and their possible role in emerging access networks are then discussed. This chapter contains more material than can be covered in the introductory course, so it is written to allow the instructor to pick and choose what sections to cover.

Chapter 4 discusses digital transmission systems and the telephone network. The first few sections deal with properties of current and emerging optical networks. The digital multiplexing hierarchy and the SONET standard are introduced. We develop the fault recovery features of SONET rings and we emphasize the capability of SONET optical networks to create arbitrary logical topologies under software control. We then introduce wavelength division multiplexing and explain how WDM optical networks share the flexible network configuration features of SONET. The design of circuit switches for traditional telephone networks and for future optical networks is discussed next. The latter sections deal with telephone networks, with a focus on the signaling system that enables telephone service and associated enhanced services, e.g., caller ID, 800-call. We consider the telephone network and the layered architecture of its signaling system. We discuss the frequency reuse concept and its application in telephone and satellite cellular networks.

Chapter 5 is the usual place to discuss data link controls. Instead of dealing immediately with this topic, we first introduce the notions of peer-to-peer protocols and service models. ARQ protocols that provide reliable transfer service are developed in detail as specific examples of peer-to-peer protocols. The detailed discussion gives the student an appreciation of what is involved in implementing a protocol. The end-to-end and hop-by-hop approaches to deploying peer-to-peer protocols are compared, and additional examples of peer-to-peer protocols are introduced for flow control and for timing recovery. We also preview the reliable stream service provided by TCP. The details of HDLC and

Ex. 1015.9

PPP data link standards are then presented. Finally we discuss the sharing of a data link by multiple packet flows and introduce the notion of multiplexing gain.

Chapter 6 deals with the transfer information across shared media, using LANs and wireless networks as specific examples. We begin with an introduction to broadcast networks and to approaches to sharing a medium. We explain the function of LANs and their placement in the OSI reference model. We consider random access as well as scheduling approaches to transferring packets across a shared medium. We examine the impact of delay-bandwidth product on performance, and we show why this dictates the evolution of Ethernet from a shared medium access technique to a switched technique. In addition to token ring and FDDI LANs, we also present a full discussion of the IEEE 802.11 wireless LAN standard. We also discuss FDMA, TDMA, and CDMA channelization approaches to sharing media and we show their application in various existing cellular radio networks. We have taken great care to make the difficult topic of CDMA accessible to the student.

Chapter 7 deals with packet switching networks. To provide a context for the chapter we begin by presenting an end-to-end view of packet transfer across the Internet. We then develop the notions of datagram and virtual-circuit packet switching, using IP and ATM as examples. We introduce basic design approaches to packet switches and routers. Shortest-path algorithms and the link state and distance vector approaches to selecting routes in a network are presented next. ATM and the concept of label switching are introduced, and the relationship between Quality-of-Service and traffic shaping, scheduling and call admission control is developed. The chapter includes a discussion of TCP and ATM congestion control.

Key Architectures and Advanced Topics

The third section shows how the fundamental networking concepts are embodied in two key network architectures, ATM and TCP/IP. The section also deals with the interworking of ATM and TCP/IP, as well as with enhancements to TCP/IP to provide secure and more responsive communications.

Chapter 8 presents a detailed discussion of TCP/IP protocols. We examine the structure of the IP layer and the details of IP addressing, routing, and fragmentation and reassembly. We discuss the motivation and present the features of IPv6. We introduce UDP, and examine in detail how TCP provides reliable stream service and flow control end-to-end across a connectionless packet network. RIP, OSPF, and BGP are introduced as protocols for synthesizing routing tables in the Internet. Multicast routing is also introduced.

Chapter 9 deals with the architecture of ATM networks. The ATM layer is explained, and Quality-of-Service and the ATM network service categories are presented. The various types of ATM adaptation layer protocols are discussed next. ATM signaling and PNNI routing are introduced.

Chapter 10 deals with the interworking of IP and ATM and with proposed enhancements to IP. We consider the various approaches for operating IP over

Ex. 1015.10

ATM networks. We then introduce Multiprotocol Label Switching which is the most promising example for operating IP over ATM and other link layer protocols. Finally we introduce RSVP, Integrated Services IP, and Differentiated Services IP which together provide mechanisms for providing Quality-of-Service over IP.

Chapter 11 provides an introduction to network security protocols. The various categories of threats that can arise in a network are used to identify various types of security requirements. Secret key and public key cryptography are introduced and their application to providing security is discussed. We develop protocols that provide security across insecure networks and we introduce protocols for establishing security associations and for managing keys. These general protocols are then related to the IP security protocols and to transport layer security protocols.

Chapter 12 deals with multimedia information and networking. We begin with an introduction to the properties of image, audio, and video signals. We discuss the various compression schemes that are applied to obtain efficient digital representations, and we describe the relevant compression standards. We then introduce the RTP protocol for transmitting real-time information across the Internet. Finally, we close the loop in the discussion of "plain old telephone service" by reviewing the various signaling protocols that are being developed to support multimedia communications in general, and IP telephony in particular, over the Internet.

The book ends with an Epilogue that discusses trends in network architecture and identifies several areas that are likely to influence the development of future networks.

Appendices

Appendix A deals with network performance models. Network performance is an integral part of network design and operation. In the text we use quantitative examples to illustrate the tradeoffs involved in various situations. We believe that an intuition for performance issues can be developed without delving into the underlying mathematics. Delay and loss performance results are introduced in the sections that deal with multiplexing, trunking, and medium access control. In these sections, the dynamics of the given problem are described and the key performance results are presented. The purpose of Appendix A is to develop the analysis of the performance models that are cited in the text. These analyses may be incorporated into more advanced courses on communication networks.

Appendix B provides an introduction to network management. The basic functions and structure of a network management system are introduced as well as the Simple Network Management Protocol (SNMP). We present the rules for describing management information, as well as the collection of objects, called Management Information Base, that are managed by SNMP. We also introduce remote monitoring (RMON) which offers extensive network diagnostic, planning, and performance information.

Ex. 1015.11

HOW TO USE THIS BOOK

The book was designed to support a variety of introductory courses on computer and communication networks. By appropriate choice of sections, the instructor can make adjustments to provide a desired focus or to account for the background of the students. Chapter 1 to Chapter 8 contain the core material (and more) that is covered in the typical introductory course on computer networks. For example, at the University of Toronto a 40 lecture-hour introductory undergraduate course in computer networks covers the following: Chapter 1 (all); Chapter 2 (all) including a series of lab exercises using sockets; Chapter 3 (sections 3.1, 3.2, 3.5, 3.6, 3.8.1 to 3.8.5); Chapter 4 (sections 4.1 to 4.3); Chapter 5 (all); Chapter 6 (sections 6.1 to 6.4, 6.6.1, 6.6.2); Chapter 7 (all); and Chapter 8 (sections 8.1 to 8.5). For courses that spend more time on the material in Chapter 8 or later, the material from Chapters 3 and 4 can be dropped altogether. The book contains enough material for a two-semester course sequence that provides an introductory course on computer networks followed by a course on emerging network protocols.

PEDAGOGICAL ELEMENTS

The book contains the following pedagogical elements:

- *Numerous Figures*. Network diagrams, time diagrams, performance graphs, state transition diagrams are essential to effectively convey concepts in networking. The 574 figures in the book are based on a set of Microsoft PowerPoint® course presentations that depend heavily on visual representation of concepts. A set of these presentation charts is available to instructors.
- *Numerous Examples*. The discussion of fundamental concepts is accompanied with examples illustrating the use of the concept in practice. Numerical examples are included in the text wherever possible.
- *Text Boxes*. Commentaries in text boxes are used to discuss network trends and interesting developments, to speculate about future developments, and to motivate new topics.
- *Problems*. The authors firmly believe that learning must involve problem solving. The book contains 589 problems. Each chapter includes problems with a range of difficulties from simple application of concepts to exploring, developing or elaborating various concepts and issues. Quantitative problems range from simple calculations to brief case studies exploring various aspects of certain algorithms, techniques, or networks. Simple programming exercises involving sockets and TCP/IP utilities are included where appropriate.
- An *Instructor's Solutions Manual* is available from McGraw-Hill.
- *Chapter Introductions*. Each chapter includes an introduction previewing the material covered in the chapter and in the context of the "big picture".

Ex. 1015.12

- *Chapter Summaries and Checklist of Important Terms.* Each chapter includes a summary that reiterates the most important concepts. A checklist of important terms will aid the student in reviewing the material.
- *References.* Each chapter includes a list of references. Given the introductory nature of the text, references concentrate on pointing to more advanced materials. Reference to appropriate Internet Engineering Taskforce (IETF) RFCs and research papers is made where appropriate, especially with more recent topics.
- *A web site.* The following Web site contains links to the on-line version of the solutions manual, the Powerpoint slides*, author information, and other related information: www.mhhe.com/leon-garcia.

ACKNOWLEDGMENTS

The material in the book was developed over many years in introductory as well as advanced courses in networking, both in regular undergraduate and graduate programs as well as in programs with an orientation towards professional practice. We acknowledge the feedback from the many students who participated in these courses and who used various versions of the manuscript. In particular we thank the students from CETYS University. We also acknowledge the input of the graduate students who served as teaching assistants in these courses, especially Dennis Chan, Yasser Rasheed, Mohamed Arad, Massoud Hashemi, Hasan Naser, and Andrew Jun.

We thank Anindo Banerjea, Raouf Boutaba, Michael Kaplan, and Gillian Woodruff for many exciting conversations on networking. Anindo and Raouf graciously provided some of the material that is presented in Chapter 2. We would also like to thank Anwar Elwalid and Debasis Mitra for their continued encouragement and interest in the book. We thank Yau-Ren Jenq for reviewing the fair queueing discussions in detail.

We are especially grateful to Irene Katzela for testing the manuscript in her courses. We also thank Ray Pickholtz for testing various versions of the text, including the beta version, and for his many valuable suggestions and his continued encouragement.

We thank the reviewers for their many useful comments on the various versions of the manuscript: Subrata Banerjee (Stevens Institute of Technology), John A. Copeland (Georgia Institute of Technology), Mario Gerla (UCLA), Rohit Goyal (Ohio State University), Gary Harkin (Montana State University), Melody Moh (San Jose State University), Kihong Park (Purdue University–West Lafayette), Raymond L. Pickholtz (The George Washington University), Chunming Qiao (SUNY Buffalo), Arunabha Sen

*The *Instructor's Solutions Manual* and the Powerpoint slides are password protected. See the website for information on how to obtain one.

(Arizona State University), Stuart Tewksbury (West Virginia University), and Zhi-li Zhang (University of Minnesota).

We would also like to acknowledge the many friends from Nortel Networks for showing us the many facets of networking. We thank Sidney Yip for opening the door to many years of interaction. We also thank Richard Vickers, Marek Wernik, and Jim Yan for many illuminating conversations over the years. We especially thank Tony Yuen for sharing his vast knowledge of the networking industry and for continuously showing how the big picture is actually bigger!

We thank Eric Munson from McGraw-Hill for persuading us to take the plunge with this project, and Betsy Jones, Executive Editor, for providing decisive support at key times. In addition we thank the production team at McGraw-Hill for their patience, ideas, and continued support, especially Kelley Butcher and Jim Labeots.

IW would like to thank to his wife Liesye for the constant encouragement and for putting up with him during the many nights and weekends spent writing the book, especially during the final stages.

Finally, ALG would like to thank his soulmate, Karen Carlyle, who went beyond the usual putting up with an author's neglect, to assuming the role of project manager, designer, transcriber and real-time editor for the book.

With the help of the many reviewers, professors, and students who have used early versions of this book we have tried to make the complex and fluid topic of network architecture as approachable, up-to-date and error-free as possible. We welcome all comments and suggestions on how to improve the text. Please contact us via the text's website with any ideas you may have.

Alberto Leon-Garcia
Indra Widjaja

Ex. 1015.14

CONTENTS

Preface vii

1 Communication Networks and Services 1

1.1 Networks and Services 2

1.2 Approaches to Network Design 9

1.2.1 Network Functions and Network Topology 9

1.2.2 Message, Packet, and Circuit Switching 14

1.2.3 Telegraph Networks and Message Switching 15

1.2.4 Telephone Networks and Circuit Switching 17

1.2.5 The Internet and Packet Switching 20

1.2.6 Discussion on Switching Approaches 29

1.3 Key Factors in Communication Network Evolution 30

1.3.1 Role of Technology 31

1.3.2 Role of Regulation 32

1.3.3 Role of the Market 34

1.3.4 Role of Standards 34

1.4 Book Overview 35

Checklist of Important Terms 38

Further Reading 38

Problems 39

2 Applications and Layered Architectures 43

2.1 Examples of Layering 44

2.2 The OSI Reference Model 50

2.2.1 Unified View of Layers, Protocols, and Services 50

2.2.2 The Seven-Layer Model 53

2.3 Overview of TCP/IP Architecture 57

2.3.1 TCP/IP Protocol: How the Layers Work Together 60

◆**2.4** The Berkeley API 64

2.4.1 Socket System Calls 67

2.4.2 Network Utility Functions 71

◆**2.5** Application Protocols and TCP/IP Utilities 80

2.5.1 Telnet 80

2.5.2 File Transfer Protocol 82

2.5.3 IP Utilities 84

Summary 86

Checklist of Important Terms 87

Further Reading 87

Problems 88

Ex. 1015.15

3 Digital Transmission Fundamentals 95

3.1 Digital Representation of Information 96
3.1.1 Binary Representations of Different Information Types 97
3.1.2 Network Requirements of Different Information Types 101
3.2 Why Digital Communications? 103
3.2.1 Basic Properties of Digital Transmission Systems 107
3.3 Characterization of Communication Channels 110
3.3.1 Frequency Domain Characterization 111
3.3.2 Time Domain Characterization 115
3.4 Fundamental Limits in Digital Transmission 116
3.4.1 The Nyquist Signaling Rate 116
3.4.2 The Shannon Channel Capacity 119
3.5 Line Coding 122
3.6 Modems and Digital Modulation 125
3.6.1 Signal Constellations and Telephone Modem Standards 131
3.7 Properties of Media and Digital Transmission Systems 133
3.7.1 Twisted Pair 135
3.7.2 Coaxial Cable 140
3.7.3 Optical Fiber 144
3.7.4 Radio Transmission 148
3.7.5 Infrared Light 152
3.8 Error Detection and Correction 153
3.8.1 Error Detection 154
3.8.2 Two-Dimensional Parity Checks 157
3.8.3 Internet Checksum 158
3.8.4 Polynomial Codes 161
3.8.5 Standardized Polynomial Codes 164
3.8.6 Error Detecting Capability of a Polynomial Code 165
◆*3.8.7 Linear Codes* 167
◆*3.8.8 Error Correction* 173
Summary 177
Checklist of Important Terms 178
Further Reading 179
Problems 180
Appendix 3A: Asynchronous Data Transmission 187
Appendix 3B: Fourier Series 190

4 Transmission Systems and the Telephone Network 192

4.1 Multiplexing 193
4.1.1 Frequency-Division Multiplexing 194
4.1.2 Time-Division Multiplexing 195
4.2 SONET 198
4.2.1 SONET Multiplexing 199
4.2.2 SONET Frame Structure 204
4.3 Wavelength-Division Multiplexing 208

Ex. 1015.16

4.4 Circuit Switches 211
4.4.1 Space-Division Switches 212
◆*4.4.2 Time-Division Switches* 215
4.5 The Telephone Network 219
4.5.1 Transmission Facilities 222
4.5.2 End-to-End Digital Services 225
4.6 Signaling 227
4.6.1 Signaling System #7 Architecture 230
4.7 Traffic and Overload Control in Telephone Networks 232
4.7.1 Concentration 232
4.7.2 Routing Control 235
4.7.3 Overload Controls 237
4.8 Cellular Telephone Networks 239
4.9 Satellite Cellular Networks 246
Summary 249
Checklist of Important Terms 250
Further Reading 251
Problems 252

5 Peer-to-Peer Protocols 262
5.1 Peer-to-Peer Protocols and Service Models 264
5.1.1 Service Models 267
5.1.2 End to End Requirements and Adaptation Functions 268
5.1.3 End to End versus Hop by Hop 271
5.2 ARQ Protocols 272
5.2.1 Stop-and-Wait ARQ 274
5.2.2 Go-Back-N ARQ 278
5.2.3 Selective Repeat ARQ 285
5.2.4 Transmission Efficiency of ARQ Protocols 289
5.3 Other Adaptation Functions 296
5.3.1 Sliding Window Flow Control 297
5.3.2 Timing Recovery for Synchronous Services 298
5.3.3 Reliable Stream Service 302
5.4 Data Link Controls 303
5.4.1 HDLC Data Link Control 303
5.4.2 Point-to-Point Protocol 311
◆**5.5** Link Sharing using Packet Multiplexers 313
5.5.1 Statistical Multiplexing 314
5.5.2 Speech Interpolation and the Multiplexing of Packetized Speech 322
Summary 326
Checklist of Important Terms 327
Further Reading 328
Problems 328

Ex. 1015.17

6 Local Area Networks and Medium Access Control Protocols 339

6.1 Multiple Access Communications 341
6.2 Local Area Networks 349
6.2.1 LAN Structure 349
6.2.2 The Medium Access Control Sublayer 351
6.2.3 The Logical Link Control Layer 352
6.3 Random Access 354
6.3.1 ALOHA 355
6.3.2 Slotted ALOHA 358
6.3.3 CSMA 359
6.3.4 CSMA-CD 361
6.4 Scheduling Approaches to Medium Access Control 366
6.4.1 Reservation Systems 366
6.4.2 Polling 368
6.4.3 Token-Passing Rings 372
6.4.4 Comparison of Scheduling Approaches to Medium Access Control 378
6.4.5 Comparison of Random Access and Scheduling Medium Access Controls 379
◆**6.5** Channelization 380
6.5.1 FDMA 380
6.5.2 TDMA 381
6.5.3 CDMA 382
6.5.4 Channelization in Telephone Cellular Networks 389
6.5.5 Performance of Channelization Techniques with Bursty Traffic 396
6.6 LAN Standards 398
6.6.1 Ethernet and IEEE 802.3 LAN Standard 399
6.6.2 Token Ring and IEEE 802.5 LAN Standard 408
6.6.3 FDDI 414
6.6.4 Wireless LANs and IEEE 802.11 Standard 417
6.7 LAN Bridges 437
6.7.1 Transparent Bridges 439
6.7.2 Source Routing Bridges 446
6.7.3 Mixed-Media Bridges 449
Summary 449
Checklist of Important Terms 451
Further Reading 452
Problems 452

7 Packet-Switching Networks 460

7.1 Network Services and Internal Network Operation 462
7.2 Packet Network Topology 466
7.3 Datagrams and Virtual Circuits 471

Ex. 1015.18

7.3.1 Structure of Switch/Router 472
7.3.2 Connectionless Packet Switching 475
7.3.3 Virtual-Circuit Packet Switching 480
7.4 Routing in Packet Networks 484
7.4.1 Routing Algorithm Classification 485
7.4.2 Routing Tables 486
7.4.3 Hierarchical Routing 488
7.4.4 Link State versus Distance Vector Routing 489
7.5 Shortest Path Algorithms 490
7.5.1 The Bellman-Ford Algorithm 491
7.5.2 Dijkstra's Algorithm 497
7.5.3 Other Routing Approaches 499
7.6 ATM Networks 503
7.7 Traffic Management and QoS 507
7.7.1 FIFO and Priority Queues 509
7.7.2 Fair Queueing 511
7.8 Congestion Control 516
7.8.1 Open-Loop Control 518
7.8.2 Closed-Loop Control 527
Summary 534
Checklist of Important Terms 535
Further Reading 536
Problems 536

8 TCP/IP 544
8.1 The TCP/IP Architecture 545
8.2 The Internet Protocols 548
8.2.1 IP Packet 548
8.2.2 IP Addressing 550
8.2.3 Subnet Addressing 552
8.2.4 IP Routing 554
8.2.5 Classless Interdomain Routing (CIDR) 556
8.2.6 Address Resolution 557
8.2.7 Reverse Address Resolution 558
8.2.8 Fragmentation and Reassembly 559
8.2.9 ICMP: Error and Control Messages 561
8.3 IPv6 561
8.3.1 Header Format 562
8.3.2 Network Addressing 563
8.3.3 Extension Headers 566
8.4 User Datagram Protocol 569
8.5 Transmission Control Protocol 570
8.5.1 TCP Reliable Stream Service 570
8.5.2 TCP Operation 571
8.5.3 TCP Protocol 575

Ex. 1015.19

8.6 DHCP and Mobile IP 587
8.6.1 Dynamic Host Configuration Protocol 588
8.6.2 Mobile IP 588
8.7 Internet Routing Protocols 590
8.7.1 Routing Information Protocol 592
8.7.2 Open Shortest Path First 594
8.7.3 Border Gateway Protocol 602
8.8 Multicast Routing 611
8.8.1 Reverse-Path Broadcasting 612
8.8.2 Internet Group Management Protocol 614
8.8.3 Reverse-Path Multicasting 615
8.8.4 Distance-Vector Multicast Routing Protocol 617
Summary 617
Checklist of Important Terms 618
Further Reading 619
Problems 620

9 ATM Networks 627

9.1 Why ATM? 628
9.2 BISDN Reference Model 629
9.3 ATM Layer 633
9.3.1 ATM Cell Header 633
9.3.2 Virtual Connections 635
9.3.3 QoS Parameters 636
9.3.4 Traffic Descriptors 638
9.3.5 ATM Service Categories 638
9.3.6 Traffic Contracts, Connection Admission Control, and Traffic Management 641
9.4 ATM Adaptation Layer 642
9.4.1 AAL1 644
9.4.2 AAL2 646
9.4.3 AAL3/4 649
9.4.4 AAL5 650
9.4.5 Signaling AAL 653
9.4.6 Applications, AALs, and ATM Service Categories 655
9.5 ATM Signaling 657
9.5.1 ATM Addressing 658
9.5.2 UNI Signaling 659
9.5.3 PNNI Signaling 662
9.6 PNNI Routing 664
Summary 668
Checklist of Important Terms 668
Further Reading 668
Problems 669

Ex. 1015.20

10 Advanced Network Architectures 675

10.1 IP Forwarding Architectures 676
10.2 Overlay Model 677
10.2.1 Classical IP Over ATM 677
10.2.2 LANE 679
10.2.3 NHRP 680
10.2.4 MPOA 682
10.3 MPLS 685
10.3.1 Fundamentals of Labels 686
10.3.2 Label Stack 688
10.3.3 VC Merging 689
10.3.4 Label Distribution Protocol 690
10.3.5 Explicit Routing for Traffic Engineering 691
10.3.6 MPLS Support for Virtual Networks 692
10.4 Integrated Services in the Internet 693
10.4.1 Guaranteed Service 694
10.4.2 Controlled-Load Service 695
10.5 RSVP 695
10.5.1 Receiver-Initiated Reservation 697
10.5.2 Reservation Merging 698
10.5.3 Reservation Styles 699
10.5.4 Soft State 700
10.5.5 RSVP Message Format 701
10.6 Differentiated Services 703
10.6.1 DS Field 705
10.6.2 Per Hop Behaviors 706
10.6.3 Traffic Conditioner 708
10.6.4 Bandwidth Broker 708
Summary 709
Checklist of Important Terms 710
Further Reading 710
Problems 711

11 Security Protocols 716

11.1 Security and Cryptographic Algorithms 717
11.1.1 Applications of Cryptography to Security 719
11.1.2 Key Distribution 724
11.2 Security Protocols 727
11.2.1 Application Scenarios 727
11.2.2 Types of Service 729
11.2.3 Setting Up a Security Association 732
11.2.4 IPSec 735
11.2.5 Secure Sockets Layer and Transport Layer Security 737
11.3 Cryptographic Algorithms 741
11.3.1 DES 741

Ex. 1015.21

11.3.2 RSA 744
Summary 746
Checklist of Important Terms 747
Further Reading 748
Problems 748

12 Multimedia Information and Networking 752
12.1 Lossless Data Compression 753
12.1.1 Huffman Codes 753
12.1.2 Run-Length Codes 758
12.1.3 Adaptive Codes 763
12.2 Digital Representation of Analog Signals 765
12.2.1 Properties of Analog Signals 766
12.2.2 Analog-to-Digital Conversion 768
12.2.3 Digital Transmission of Analog Signals 770
◆*12.2.4 SNR Performance of Quantizers* 775
12.3 Techniques for Increasing Compression 776
12.3.1 Predictive Coding 778
12.3.2 Transform Coding 780
12.3.3 Subband Coding 781
12.3.4 Discrete Cosine Transform Coding 782
12.3.5 The JPEG Image-Coding Standard 784
12.3.6 Compression of Video Signals 786
12.3.7 The MPEG Video Coding Standards 789
12.3.8 MPEG Multiplexing 792
12.4 The Real-Time Transport Protocol 793
12.4.1 RTP Scenarios and Terminology 794
12.4.2 RTP Packet Format 795
12.4.3 RTP Control Protocol (RTCP) 796
12.5 Session Control Protocols 798
12.5.1 Session Initiation Protocol 798
12.5.2 H.323 Multimedia Communications Systems 800
12.5.3 Media Gateway Control Protocols 802
Summary 803
Checklist of Important Terms 803
Further Reading 804
Problems 804

Epilogue 812

Appendices
A Delay and Loss Performance 815
A.1 Delay Analysis and Little's Formula 815
A.1.1 Arrival Rates and Traffic Load Definitions 816

Ex. 1015.22

A.1.2 Little's Formula 818
A.2 Basic Queueing Models 821
A.2.1 Arrival Processes 821
A.2.2 Service Times 822
A.2.3 Queueing System Classification 823
A.3 M/M/1: A Basic Multiplexer Model 825
A.3.1 M/M/1 Steady State Probabilities and the Notion of Stability 826
A.3.2 Effect of Scale on Performance 829
A.3.3 Average Packet Delay in a Network 829
A.4 The M/G/1 model 830
A.4.1 Service Time Variability and Delay 831
A.4.2 Priority Queueing Systems 832
A.4.3 Vacation Models and Multiplexer Performance 833
A.5 Erlang B Formula: M/M/c/c System 833
Further Reading 835

B Network Management 836
B.1 Network Management Overview 837
B.2 Simple Network Management Protocol (SNMP) 839
B.3 Structure of Management Information 842
B.4 Management Information Base 844
B.5 Remote Network Monitoring 845
Further Reading 845

Index 847

Ex. 1015.23

CHAPTER 3

Digital Transmission Fundamentals

In Chapter 1 we saw that the development of the early network architectures was motivated by very specific applications. Telegraphy was developed specifically for the transfer of text messages. We saw that Morse and Baudot pioneered the use of binary representations for the transfer of text and that digital transmission systems were developed for the transfer of the resulting binary information streams. Later telephony was developed for the transfer of analog voice information. The invention of **Pulse Code Modulation (PCM)** paved the way for voice to be transmitted over digital transmission networks. In the same way that the Morse and Baudot codes standardized the transfer of text, PCM standardized the transfer of voice in terms of 0s and 1s. We are currently undergoing another major transition from analog to digital transmission technology, namely, the transition from analog television systems to entirely digital television systems. When this transition is complete, all major forms of information will be represented in digital form. This change will open the way for the deployment of digital networks that can transfer the information for all the major types of information services.

Digital transmission is the core technology that enables the integration of services in a network. In this chapter we present the fundamental concepts concerning digital transmission. The chapter is organized into the following sections:

1. *Digital representation of information.* The basic properties of text, image, voice, audio and video information are presented. These information types can be viewed as flows that traverse the network. We examine the types of information flows that are generated by the various types of information, and we consider the requirements that they place on the networks that carry them.
2. *Why digital transmission?* We explain the advantages of digital transmission over analog transmission. We develop the basic abstraction of a transmission link as a pipe that transfers bits for the data link layer. We present the key

Ex. 1015.24

parameters that determine the transmission capacity of a physical medium. We also indicate where various media are used in digital transmission systems. This section is a summary of the following two sections.[1]

3. *Characterization of communication channels.* We discuss communication channels in terms of their ability to transmit pulse and sinusoidal signals. We introduce the concept of bandwidth as a measure of a channel's ability to transmit pulse information.
4. *Fundamental limits of digital transmission.* We discuss binary and multilevel digital transmission systems, and we develop fundamental limits on the bit rate that can be obtained over a channel.
5. *Line coding.* We introduce various formats for transmission binary information and discuss the criteria for selecting an appropriate line code.
6. *Modems and digital modulation.* We discuss digital transmission systems that use sinusoidal signals, and we explain existing telephone modem standards.
7. *Properties of transmission media.* We discuss copper wire, radio, and optical fiber systems and their role in access and backbone digital networks. Examples from various physical layer standards are provided.
8. *Error detection and correction.* We present coding techniques that can be used to detect and correct errors that may occur during digital transmission. These coding techniques form the basis for protocols that provide reliable transfer of information.

3.1 DIGITAL REPRESENTATION OF INFORMATION

Networks are driven by the applications they support and must therefore be designed to accommodate the requirements imposed by the information types used in the applications. These information types include text, speech, audio, data, images, and video. In this section we are concerned with (1) the properties of these information types and (2) the requirements that the digital representation of these information types impose on the network. We are concerned in particular with the volume of binary information that needs to be transferred for each information type. The remainder of the chapter is concerned with techniques for carrying out the transfer of such information over various types of communications systems.

It is useful at this point to identify which layers in the OSI reference model we are dealing with. This section examines the information in its original form in the application. The application generates blocks or streams of information that all the layers in the protocol stack must handle. The remainder of the chapter

[1]This arrangement allows sections 3.3 and 3.4 to be skipped in courses that are under tight time constraints. Nevertheless, we believe that the remaining sections make the topic of digital transmission quite accessible, so we encourage all students to read the remainder of the chapter independently.

deals with the physical layer, which handles the transfer of raw bits. A useful analogy is to view the information generated by the application as a flow that needs to be carried by the network. The digital transmission systems at the physical layer are the pipes that carry the information flows.

3.1.1 Binary Representations of Different Information Types

Information can be categorized into two broad categories: information that occurs naturally in the form of a single *block* and *stream* information that is produced continuously by a source of information. Table 3.1 gives examples of **block-oriented information**, which include data files, black-and-white documents, and pictures. Table 3.2 gives examples of stream information such as audio and video.

The most common examples of block information are files that contain text, numerical, or graphical information. We routinely deal with these types of information when we send e-mail and when we retrieve documents. These blocks of information can range from a few bytes to several hundred kilobytes and occasionally several megabytes. The normal form in which these files occur can contain a fair amount of statistical redundancy. For example, certain characters and patterns such as *e* and *the*, occur very frequently. **Data compression** utilities such as compress, zip, and other variations exploit these redundancies to encode the original information into files that require less disk storage space.[2] Some modem standards also apply these data compression schemes to the information prior to transmission. The **compression ratio** is defined as the ratio of the number of bits in the original file to the number of bits in the compressed file. Typically

Information type	Data compression technique	Format	Uncompressed	Compressed (compression ratio)	Applications
Text files	Compress, zip, and variations	ASCII	Kbytes to Mbytes	(2–6)	Disk storage, modem transmission
Scanned black-and-white documents	CCITT Group 3 facsimile standard	A4 page @ 200 × 200 pixels/inch and options	256 Kbytes	15–54 Kbytes (1-D) 5–35 Kbytes (2-D) (5–50)	Facsimile transmission, document storage
Color images	JPEG	8-×-10 inch photo scanned @ 400 pixels/inch	38.4 Mbytes	1.2–8 Mbytes (5–30)	Image storage or transmission

TABLE 3.1 Block-oriented information

[2]The details of the data compression techniques discussed in this section are found in Chapter 12.

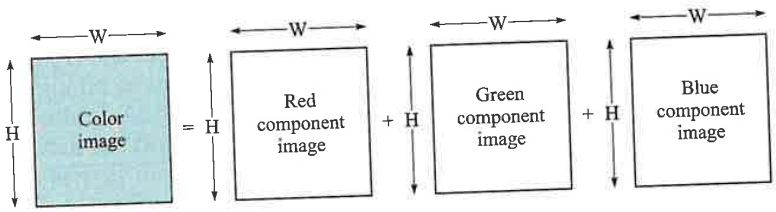
Ex. 1015.26

the compression ratio for these types of information is two or more, thus providing an apparent doubling or more of the transmission speed or storage capacity.

A facsimile system scans a black-and-white document into an array of dots that are either white or black. The CCITT Group 3 facsimile standards provide for resolutions of 200, 300, or 400 dots per horizontal inch and 100, 200, or 400 vertical lines per inch. For example, a standard A4 page at 200 × 100 pixels/inch (slightly bigger than 8.5 × 11 inches) produces 256 kilobytes prior to compression. At a speed of 28.8 kbps, such an uncompressed page would require more than 1 minute to transmit. Existing fax compression algorithms can reduce this transmission time typically by a factor of 8 to 16.

An individual color image produces a huge number of bits. A *pixel* is defined as a single dot in a digitized image. For example, an 8-×-10-inch picture scanned at a resolution of 400 × 400 pixels per square inch yields 400 × 400 × 8 × 10 = 12.8 million pixels; see Table 3.1. A color image is decomposed into red, green and blue subimages as shown in Figure 3.1. Normally eight bits are used to represent each of the red, green, and blue color components, resulting in a total of 12.8 megapixels × 3 bytes/pixel = 38.4 megabytes. At a speed of 28.8 kbps, this image would require about 3 hours to transmit! Clearly, data compression methods are required to reduce these transmission times.

The *Graphics Interchange Format (GIF)* takes image data, in binary form, and applies lossless data compression. **Lossless data compression** schemes produce a compressed file from which the original data can be recovered *exactly*. (Facsimile and file compression utilities also use lossless data compression.) However, lossless data compression schemes are limited in the compression rates they can achieve. For this reason, GIF is used mainly for simple images such as line drawings and images containing simple geometrical shapes. On the other hand, **lossy data compression** schems produce a compressed file from which only an *approximation* to the original information can be recovered. Much higher compression schemes are possible. In the case of images, lossy compression is acceptable as long as there is little or no visible degradation in image quality. The *Joint Photographic Experts Group (JPEG)* standard provides a lossy compression algorithm that can be adjusted to balance image quality versus file size. For example, JPEG can typically produce a high-quality reproduction with a compression ratio of about 15. Combined with the fastest telephone modems, say



Total bits before compression = 3 × H × W pixels × B bits/pixel = 3 HWB

FIGURE 3.1 Color image had three components

Ex. 1015.27

56 kbps, this compression ratio reduces the transmission time of the image in Table 3.1 to several minutes. Clearly in the case of images, we either make do with lower resolution and/or lower quality images or we procure higher speed communications.

Analog information such as voice, music, or video occurs in a steady stream. Table 3.2 lists the properties of this type of information. In the case of a voice or music signal, the sound, which consists of variations in air pressure, is converted into a voltage that varies continuously with time.

The first step in digitizing such a signal is to sample the values of the signal every T seconds as shown in Figure 3.2a. Clearly, the value of T that is used depends on how fast the signal varies with time. For example, for a PCM telephone-quality voice, the signal is sampled at a rate of 8000 samples/second, that is, $T = 1/8000 = 125$ microseconds as shown in Figure 3.2a. The precision of the sample measurements determines the bit rate as well as the quality of the approximation. For example, Figure 3.2b shows how each of the signal samples can be approximated by one of the eight levels. Each level can then be represented by a three-bit number. In the case of telephone systems, the samples are represented by 8 bits in resolution, resulting in a bit rate of 8000 samples/second $\times$ 8 bits/sample = 64 kbps. The bit rate can be reduced by reducing the resolution, but doing so results in a less accurate reproduction of the original signal.

Information type	Compression technique	Format	Uncompressed	Compressed	Applications
Voice	PCM	4 kHz voice	64 kbps	64 kbps	Digital telephony
Voice	ADPCM (+ silence detection)	4 kHz voice	64 kbps	32 kbps (16 kbps)	Digital telephony, voice mail
Voice	Residual-excited linear prediction	4 kHz voice	64 kbps	8–16 kbps	Digital cellular telephony
Audio	MPEG audio MP3 compression	16–24 kHz audio	512–748 kbps	32–384 kbps	MPEG audio
Video	H.261 coding	176 × 144 or 352 × 288 frames @ 10–30 frames/second	2–36.5 Mbps	64 kbps–1.544 Mbps	Video conferencing
Video	MPEG-2	720 × 480 frames @ 30 frames/second	249 Mbps	2–6 Mbps	Full-motion broadcast video
Video	MPEG-2	1920 × 1080 frames @ 30 frames/second	1.6 Gbps	19–38 Mbps	High-definition television

TABLE 3.2 Properties of analog and video information

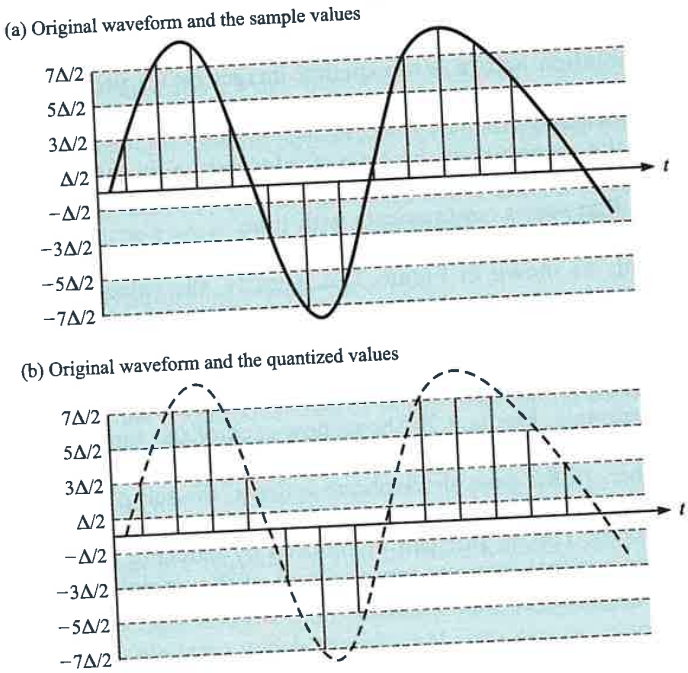


FIGURE 3.2 Sampling of a speech signal

The high cost of transmission in certain situations, for example, undersea cable, satellite, and wireless systems, has led to the development of more complex algorithms for reducing the bit rate while maintaining a telephone-quality voice signal. Differential PCM (DPCM) encodes the difference between successive samples. Adaptive DPCM (ADPCM) adapts to variations in voice-signal level. Linear predictive methods adapt to the type of sound. These systems can reduce the bit rate of telephone-quality voice to the range 8 to 32 kbps. Despite the fact that they are "lossy," these schemes achieve compression and high quality due to the imperceptibility of the approximation errors that are introduced.

Music signals vary much more rapidly than voice signals do. Thus, for example, audio compact disc (CD) systems sample the music signals at 44 kilosamples/second at a resolution of 16 bits. For a stereo music system the result is a bit rate of 44,000 samples/second × 16 bits/sample × 2 channels = 1.4 megabits/second. One hour of music will then produce 317 Mbytes of information. The subband coding technique used in the MPEG audio standard can reduce this bit rate by a factor of 14 kbps to about 100 kbps.

Video signals ("moving pictures" or "flicks") can be viewed as a succession of pictures that is fast enough to give the human eye the appearance of continuous motion. If there is very little motion, such as a close-up view of a face in a videoconference, then the system needs to transmit only the differences between

successive pictures. Typical videoconferencing systems operate with frames of 176 × 144 pixels at 10 to 36 frames/second as shown in Figure 3.3a. The color of each pixel is initially represented by 24 bits, that is, 8 bits per color component. When compressed, these videoconferencing signals produce bit rates in the range of several hundred kilobits/second as shown in Table 3.2.

Broadcast television requires greater resolution (720 × 480 pixels per frame) than video conferencing requires, as shown in Figure 3.3b, and can contain a high degree of motion. The MPEG-2 coding system can achieve a reduction from the uncompressed bit rate of 249 Mbps to the range of 2 to 6 Mbps. The recently approved Advanced Television Systems Committee (ATSC) U.S. standard for high-definition television applies the MPEG-2 coding system in a system that operates with more detailed 1920 × 1080 pixel frames at 30 frames/second as shown in Figure 3.3c. The 16:9 aspect ratio of the frame gives a more theaterlike experience; ordinary television has a 4:3 aspect ratio. The uncompressed bit rate is 1.6 gigabits/second. The MPEG-2 coding can reduce this to 19 to 38 Mbps, which can be supported by digital transmission over terrestrial broadcast and cable television systems.

3.1.2 Network Requirements of Different Information Types

In the preceding discussion we focused primarily on the amount of information required to represent different types of information, that is, bits per file for blocks of information and bits per second for streams of information. To handle a particular type of information, a network or transmission system must be capable of transferring the associated *volume of information*. The various types

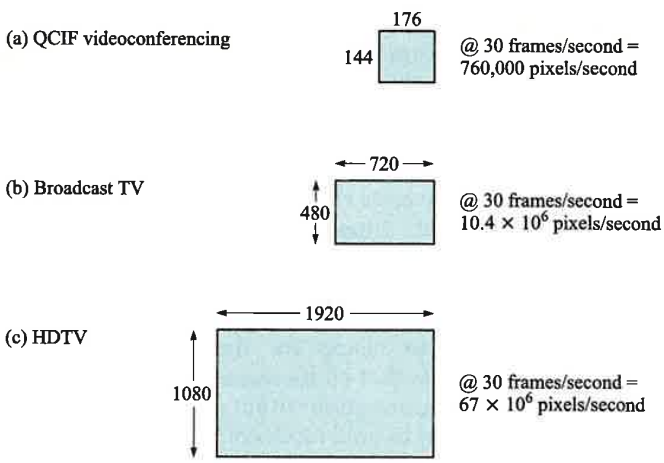


FIGURE 3.3 Video image pixel rates

of information also impose additional requirements on the network or transmission system. The *accuracy* of the information delivered and the *timeliness* of the delivery are of particular interest.

This chapter shows that digital transmission systems have some small, but nonzero, rate at which they introduce errors into the delivered information. Different types of information have different degrees of tolerance to such **transmission errors**. For example, data files in general cannot tolerate any errors. The transfer of data files therefore involves the use of error-detection schemes that identify transmissions that contain errors and initiate error-recovery procedures. Other types of information, such as facsimile, can tolerate some errors. Typically, as the compression ratio of a scheme increases, the importance of the information conveyed by every bit in the compressed sequence increases accordingly. Thus errors introduced in the compressed sequence can lead to high or even catastrophic distortion. For example, PCM speech, which is not highly compressed, can tolerate fairly high bit error rates. On the other hand, highly compressed speech, for example, using residual-excited linear predictive coding, will be quite vulnerable to errors and therefore necessitates the use of error-correction coding. Error-detection and -correction schemes are discussed in the last section of this chapter.

Certain applications have timeliness requirements associated with the delivery of information. For example, a request for a block of information may involve a presentation deadline. The network may therefore be required to deliver the block within a certain maximum **delay**. The time to deliver a block of L bits of information includes the propagation delay as well as the block transmission time: $t_{prop} + L/R$. The propagation delay $t_{prop} = d/c$ where d is that distance that the information has to travel and c is the speed of light in the transmission medium. Clearly, the designer cannot change the speed of light, but the distance that the information has to travel can be controlled through the placement of the file servers. The time to transmit the file can be reduced by increasing the transmission bit rate R.

Stream information can be viewed as a sequence of blocks of information as shown in Figure 3.4a. An application may impose a maximum delay on *every* block of information within the stream. For example, real-time communications between people requires a maximum delay of about 250 ms to ensure interactivity close to that of normal conversation.

As a stream of blocks traverses the network, the spacing between the information blocks can be altered. **Jitter** is defined as the variation in the delay between consecutive blocks. Figure 3.4b shows that variable delays incurred while traversing a network can introduce jitter through "bunching up." In stream applications the decoder that plays back the information must be fed a steady stream of information blocks for the system to operate correctly. Typically, the decoder assumes that all blocks will arrive within some maximum delay, and it plays back the information within such a delay, as shown in Figure 3.4c. The receiver uses a buffer to hold blocks of information until their playback time. The jitter is a measure of how much the blocks of information will tend to bunch up and hence of the buffer size that will be required to hold the blocks.

Ex. 1015.31

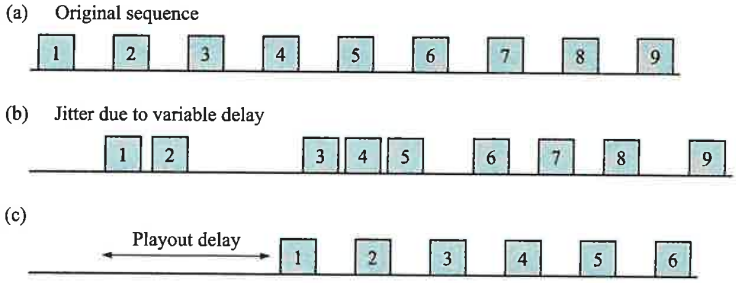


FIGURE 3.4 Temporal impairments for stream information

In the remainder of this chapter we are concerned only with the bit rate and the transmission error properties of digital transmission systems. We also consider the problem of dealing with transmission errors. In Chapter 5 we return to the discussion of the adaptation techniques that can be used to help an application deal with the various types of impairments incurred during transfer through a network.

3.2 WHY DIGITAL COMMUNICATIONS?[3]

A transmission system makes use of a physical **transmission medium** or **channel** that allows the propagation of energy in the form of pulses or variations in voltage, current, or light intensity as shown in Figure 3.5. In analog communications the objective is to transmit a waveform, which is a function that varies continuously with time, as shown in Figure 3.6a. For example, the electrical signal coming out of a microphone corresponds to the variation in air pressure corresponding to sound. This function of time must be reproduced exactly at the output of the analog communication system. In practice, communications channels do not satisfy this condition, so some degree of distortion is unavoidable.

In digital transmission the objective is to transmit a given symbol that is selected from some finite set of possibilities. For example, in binary digital

FIGURE 3.5 General transmission system

[3]This section summarizes the main results of sections 3.3 and 3.4, allowing these two sections to be skipped if necessary.

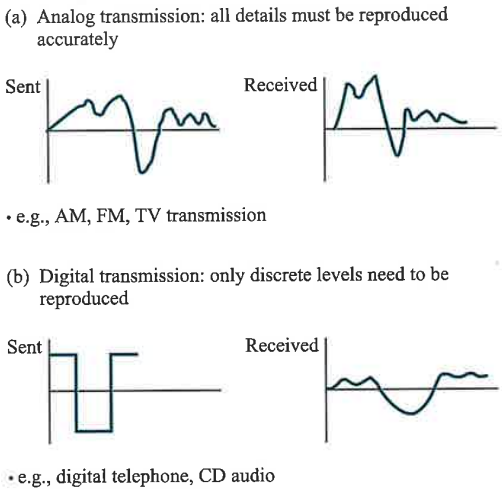


FIGURE 3.6 Analog versus digital transmission

transmission the objective is to transmit either a 0 or a 1. This can be done, for instance, by transmitting positive voltage for a certain period of time to convey a 1 or a negative voltage to convey a 0, as shown in Figure 3.6b. The task of the receiver is to determine the input symbol with high probability. The positive or negative pulses that were transmitted for the given symbols can undergo a great degree of distortion. Where signaling uses positive or negative voltages, the system will operate correctly as long as the receiver can determine whether the original voltage was positive or negative.

The cost advantages of digital transmission over analog transmission become apparent when transmitting over a long distance. Consider, for example, a system that involves transmission over a pair of copper wires. As the length of the pair of wires increases, the signal at the output is attenuated and the original shape of the signal is increasingly distorted. In addition, interference from extraneous sources, such as radiation from car ignitions and power lines, as well as noise inherent in electronic systems result in the addition of random noise to the transmitted signal. To transmit over long distances, it is necessary to introduce repeaters periodically to regenerate the signal, as shown in Figure 3.7. Such signal regeneration is fundamentally different for analog and digitial transmissions.

In an analog communication system, the task of the repeater is to regenerate a signal that resembles as closely as possible the signal at the input of the repeater segment. Figure 3.8 shows the basic functions carried out by the repeater. The

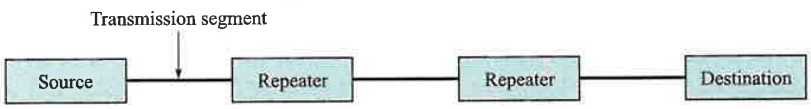


FIGURE 3.7 Typical long-distance link

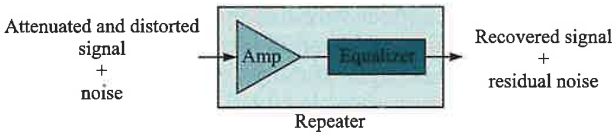


FIGURE 3.8 An analog repeater

input to the repeater is an attenuated and distorted version of the original transmitted signal plus the random noise added in the segment. First the repeater deals with the attenuation by amplifying the received signal. To do so the repeater multiplies the signal by a factor that is the reciprocal of the attenuation a. The resulting signal is still distorted by the channel.

The repeater next uses a device called an **equalizer** in an attempt to eliminate the distortion. You will see in the next section that the source of the distortion in the signal shape has two primary causes. The first cause is that different frequency components of the signal are attenuated differently.[4] In general, high-frequency components are attenuated more than low-frequency components. The equalizer compensates for this situation by amplifying different frequency components by different amounts. The second cause is that different frequency components of a signal are delayed by different amounts as they propagate through the channel. The equalizer attempts to provide differential delays to realign the frequency components. In practice it is very difficult to carry out the two functions of the equalizer. For the sake of argument, suppose that the equalization is perfect. The output of the repeater then consists of the original signal plus the noise.

In the case of analog signals, the repeater is limited in what it can do to deal with noise. If it is known that the original signal does not have components outside a certain frequency band, then the repeater can remove noise components that are outside the signal band. However, the noise within the signal band cannot be reduced and consequently the signal that is finally recovered by the repeater will contain some noise. The repeater then proceeds to send the recovered signal over the next transmission segment.

The effect on signal quality after multiple analog repeaters is similar to that in repeated recordings using analog audiocassette tapes or VCR tapes. The first time a signal is recorded, a certain amount of noise, which is audible as hiss, is introduced. Each additional recording adds more noise. After a large number of recordings, the signal quality degrades considerably.[5] A similar effect occurs in the transmission of analog signals over multiple repeater segments.

[4]Periodic signals can be represented as a sum of sinusoidal signals using Fourier series. Each sinusoidal signal has a distinct frequency. We refer to the sinusoidal signals as the "frequency components" of the original signal. (Fourier series are reviewed in Appendix 3B.)

[5]Digital recording techniques will be introduced in consumer products in the near future, and so this example will become obsolete! Another example involves noting the degradation in image quality as a photocopy of a photocopy is made.

Next consider the same copper wire transmission system for digital communications. Suppose that a string of 0s and 1s is conveyed by a sequence of positive and negative voltages. As the length of the pair of wires increases, the pulses are increasingly distorted and more noise is added. A digital repeater is required as shown in Figure 3.7. The sole objective of the repeater is to determine with high probability the original binary stream. The repeater also uses an equalizer to compensate for the distortion introduced by the channel. However, the repeater does not need to completely regenerate the original shape of the transmitted signal. It only needs to determine whether the original pulse was positive or negative. To do so, the repeater is organized in the manner shown in Figure 3.9.

A timing recovery circuit keeps track of the intervals that define each pulse. The decision circuit then samples the signal at the midpoint of each interval to determine the polarity of the pulse. In a property designed system, in the absence of noise, the original symbol would be recovered every time, and consequently the binary stream would be regenerated exactly over any number of repeaters and hence over arbitrarily long distances. However, noise is unavoidable, which implies that errors will occur from time to time. An error occurs when the noise signal is sufficiently large to change the polarity of the original signal at the sampling point. Digital transmission systems are designed for very low bit error rates, for example, 10^{-7}, 10^{-9}, or even 10^{-12}, which corresponds to one error in every trillion bits!

The impact on signal quality in multiple digital repeaters is similar to the digital recording of music where the signal is stored as a file of binary information. We can copy the file digitally any number of times with extremely small probabilities of errors being introduced in the process. In effect, the quality of the sound is unaffected by the number of times the file is copied.

The preceding discussion shows that digital transmission has superior performance over analog transmission. Digital repeaters eliminate the accumulation of noise that takes place in analog systems and provide for long-distance transmission that is nearly independent of distance. Digital transmission systems can operate with lower signal levels or with greater distances between repeaters than analog systems can. This factor translates into lower overall system cost and was the original motivation for the introduction of digital transmission.

Over time, other benefits of digital transmission have become more prominent. Networks based on digital transmission are capable of handling *any* type of information that can be represented in digital form. Thus digital networks are

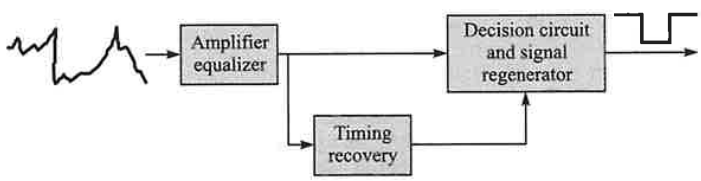


FIGURE 3.9 A digital repeater

suitable for handling many types of services. Digital transmission also allows networks to exploit the advances in digital computer technology to increase not only the volume of information that can be transmitted but also the types of processing that can be carried out within the network, that is, error correction, data encryption, and the various types of network protocol processing that are the subject of this book.

3.2.1 Basic Properties of Digital Transmission Systems

The purpose of a **digital transmission** system is to transfer a sequence of 0s and 1s from a transmitter (on the left end) to a receiver (on the right) as shown in Figure 3.10. We are particularly interested in the **bit rate** or transmission speed as measured in bits/second. The bit rate R can be viewed as the cross-section of the information pipe that connects the transmitter to the receiver.

The transmission system uses pulses or sinusoids to transmit binary information over a physical transmission medium. A fundamental question in digital transmission is *how fast* can bits be transmitted *reliably* over a given medium. This capability is clearly affected by several factors including:

- The amount of *energy* put into transmitting each signal.
- The *distance* that the signal has to traverse (because the energy is dissipated and dispersed as it travels along the medium).
- The amount of *noise* that the receiver needs to contend with.
- The *bandwidth* of the transmission medium, which we explain below.

A transmission channel can be characterized by its effect on tones of various frequencies. A (sinusoidal) tone of a given frequency f is applied at the input, and the tone at the output of the channel is measured. The ability of the channel to transfer a tone of the frequency f is given by the **amplitude-response function** $A(f)$, which is defined as the ratio of the amplitude of the output tone divided by the amplitude of the input tone. Figure 3.11 shows the typical amplitude-response functions of a low-pass channel and its idealized counterpart. The **bandwidth** of a channel is defined as the range of frequencies that is passed by a channel. Our first major result in the following sections will be to show that the rate at which pulses can be transmitted over a channel is proportional to the bandwidth. In essence, we will show that if a channel has bandwidth W, then the narrowest pulse that can be transmitted over the channel has width $\tau = 1/2W$ seconds. *Thus the fastest rate at which pulses can be transmitted into the channel is given by the Nyquist rate:* $r_{max} = 2W$ pulses/second.

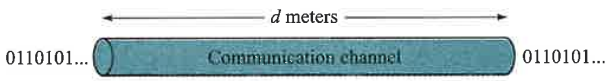


FIGURE 3.10 A digital transmission system

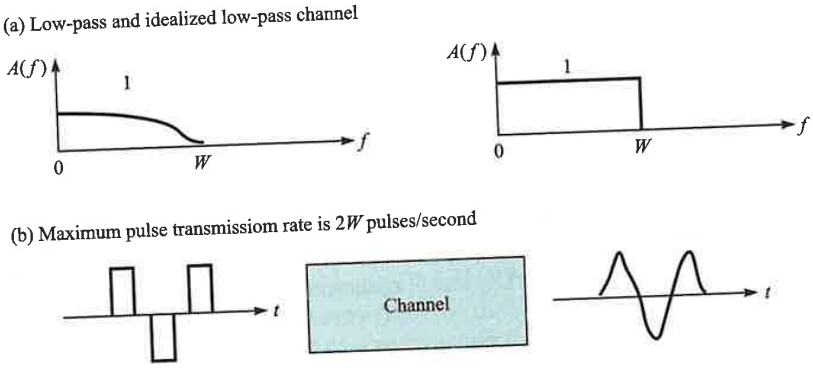


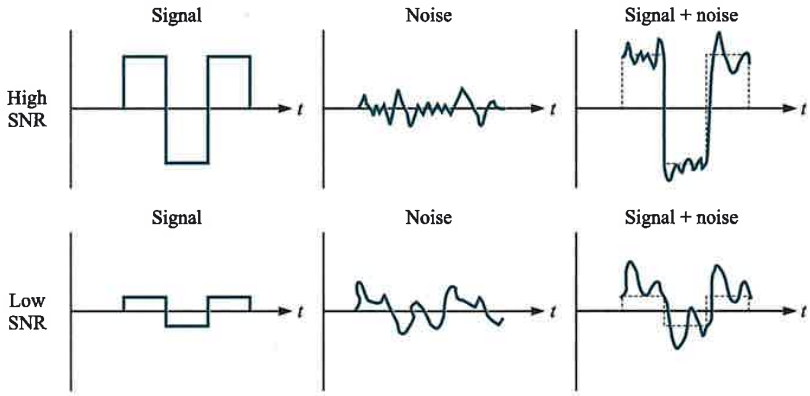
FIGURE 3.11 Typical amplitude-response functions

We can transmit binary information by sending a pulse with amplitude $+A$ to send a 1 bit and $-A$ to send a 0 bit. Each pulse transmits one bit of information, so this system then has a bit rate of $2W$ pulses/second * 1 bit/pulse = $2W$ bps. We can increase the bit rate by sending pulses with more levels. For example, if pulses can take on amplitudes from the set $\{-A, -A/3, +A/3, +A\}$ to transmit the pairs of bits {00, 01, 10, 11}, then each pulse conveys two bits of information and the bit rate is $4W$ bps. In general, if we use $M = 2^m$ levels, say, $\{-A, -(M-3)A/(M-1), \ldots, +(M-3)A/(M-1), +A\}$, then the bit rate will be $2Wm$ bps. *Thus, if we use* ***multilevel transmission*** *pulses that can take on* $M = 2^m$ *amplitude levels, we can transmit at a bit rate of* $2Wm$ *bits/second:*

$$R = 2W \text{ pulses/second} * m \text{ bits/pulse} = 2Wm \text{ bits/second}$$

In the absence of noise, the bit rate can be increased without limit by increasing the number of signal levels M. However, noise is an impairment encountered in all communication channels. Noise consists of extraneous signals that are added to the desired signal at the input to the receiver. Figure 3.12 gives two examples where the desired signal is a square wave and where noise is added to the signal. In the first example the amplitude of the noise is less than that of the desired signal, and so the desired signal is discernable even after the noise has been added. In the second example the noise amplitude is greater than that of the desired signal, which is now more difficult to discern. The **signal-to-noise ratio (SNR)**, defined in Figure 3.12, measures the relative amplitudes of the desired signal and the noise. The SNR is usually stated in decibels (dB). Returning to multilevel transmission, suppose we increase the number of levels while keeping the maximum signal levels $\pm A$ fixed. Each increase in number of signal levels requires a reduction in the spacing between levels. At some point these reductions will imply significant increases in the probability of detection errors as the noise will be more likely to convert the transmitted signal level into other signal levels. *Thus the presence of noise limits the reliability with which the receiver can correctly determine the information that was transmitted.*

Ex. 1015.37



$$\text{SNR} = \frac{\text{Average signal power}}{\text{Average noise power}}$$

$$\text{SNR (dB)} = 10 \log_{10} \text{SNR}$$

FIGURE 3.12 Signal-to-noise ratio

The **channel capacity** of a transmission system is the maximum rate at which bits can be transferred reliably. We have seen above that the bit rate and reliability of a transmission system are affected by the channel bandwidth, the signal energy or power, and the noise power. Shannon derived an expression for channel capacity. He also showed that reliable communication is not possible at rates above this capacity. The *Shannon channel capacity* is given by the following formula:

$$C = W \log_2(1 + \text{SNR})\text{bits/second}$$

As an example, consider telephone modem speeds. A modem operating over a telephone line has a maximum useful bandwidth of about 3400 Hz. If we assume an SNR of 40 dB, which is slightly over the maximum possible in a telephone line, Shannon's formula gives a channel capacity of 44.8 kbps. The V.90 modems that were introduced in 1998 operate at a rate of 56 kbps, well in excess of the Shannon bound! How can this be? The explanation is given on the next page.

Table 3.3 shows the bit rates that are provided by current digital transmission systems over various media. Twisted pairs present a wide set of options in telephone access networks and in Ethernet LANs. Cable television modems provide high speed transmission over coaxial cable. Wireless links also provide some options in access networks and LANs. Optical fiber transmission provides the high bandwidths required in backbone networks. Dense wavelength division multiplexing systems will provide huge bandwidths and will profoundly affect network design.

SHANNON CHANNEL CAPACITY AND THE 56KBPS MODEM

The Shannon channel capacity for a telephone channel gives a maximum possible bit rate of 44.8 kbps at 40 dB SNR. How is it that the V.90 modems achieve rates of 56 kbps? In fact, a look at the fine print shows that the bit rate is 33.6 kbps inbound into the network. The modem signal must undergo an analog-to-digital conversion when it is converted to PCM at the entrance to the telephone network. This step introduces the PCM approximation error or noise. At the maximum allowable signal level, we have a maximum possible SNR of 39 dB, so the 56 kbps is not attainable in the inbound direction, and hence the inbound operation is at 33.6 kbps. In the direction from the Internet server provider (ISP) to the user, the signal from the ISP is already digital and so it does not need to undergo analog-to-digital conversion. Hence the quantization noise is not introduced, a higher SNR is possible, and speeds approaching 56 kbps can be achieved from the network to the user.

Digital transmission system	Bit rate	Observations
Telephone twisted pair	33.6 kbps	4 kHz telephone channel
Ethernet over twisted pair	10 Mbps	100 meters over unshielded twisted pair
Fast Ethernet over twisted pair	100 Mbps	100 meters using several arrangements of unshielded twisted pair
Cable modem	500 kbps to 4 Mbps	Shared CATV return channel
ADSL over twisted pair	64–640 kbps inbound 1.536–6.144 Mbps outbound	Uses higher frequency band and coexists with conventional analog telephone signal, which occupies 0–4 kHz band
Radio LAN in 2.4 GHz band	2 Mbps	IEEE 802.11 wireless LAN
Digital radio in 28 GHz band	1.5–45 Mbps	5 km multipoint radio link
Optical fiber transmission system	2.4–9.6 Gbps	Transmission using one wavelength
Optical fiber transmission system	1600 Gbps and higher	Multiple simultaneous wavelengths using wavelength division multiplexing

TABLE 3.3 Bit rates of digital transmission systems

3.3 CHARACTERIZATION OF COMMUNICATION CHANNELS

A communication channel is a system consisting of a physical medium and associated electronic and/or optical equipment that can be used for the transmission of information. Commonly used physical media are copper wires, coaxial cable, radio, and optical fiber. Communications channels can be used for the transmission of either digital or analog information. Digital transmission involves the transmission of a sequence of pulses that is determined by a corre-

Ex. 1015.39

sponding digital sequence, typically a series of binary 0s and 1s. Analog transmission involves the transmission of waveforms that correspond to some analog signal, for example, audio from a microphone or video from a television camera. Communication channels can be characterized in two principal ways: frequency domain and time domain.

3.3.1 Frequency Domain Characterization

Figure 3.13 shows the approach used in characterizing a channel in the frequency domain. A sinusoidal signal $x(t) = \cos(2\pi ft)$ that oscillates at a frequency of f cycles/second (Hertz) is applied to a channel. The channel output $y(t)$ usually consists of a sinusoidal signal of the same frequency but of different amplitude and phase:[6]

$$y(t) = A(f)\cos(2\pi ft + \varphi(f)) = A(f)\cos(2\pi f(t - \tau(f))). \tag{1}$$

The channel is characterized by its effects on the input sinusoidal signal. The first effect involves an attenuation of the sinusoidal signal. This effect is characteized by the **amplitude-response** function $A(f)$, which is the ratio of the output amplitude to the input amplitude of the sinusoids at frequency f. The second effect is a **phase shift** $\varphi(f)$ in the output sinusoid relative to the input sinusoid. In general, both the amplitude response and the phase shift depend on the frequency f of the sinusoid. You will see later that for various communication channels the attenuation increases with f. Equation 1 also shows that the output $y(t)$ can be viewed as the input attenuated by $A(f)$ and delayed by $\tau(f)$.

The frequency-domain characterization of a channel involves varying the frequency f of the input sinusoid to evaluate $A(f)$ and $\varphi(f)$. Figure 3.14 shows the amplitude-response and phase-shift functions for a "low pass" channel. In this channel very low frequencies are passed, but very high frequencies are essentially eliminated. In addition, frequency components at low frequencies are not phase shifted, but very high frequencies are shifted by 90 degrees.

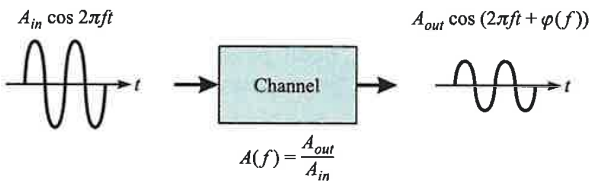


FIGURE 3.13 Channel characterization—frequency domain

[6]The statement applies to channels that are "linear." For such channels the output signal corresponding to a sum of input signals, say, $x_1(t) + x_2(t)$, is equal to the sum of the outputs that would have been obtained for each individual input; that is, $y(t) = y_1(t) + y_2(t)$.

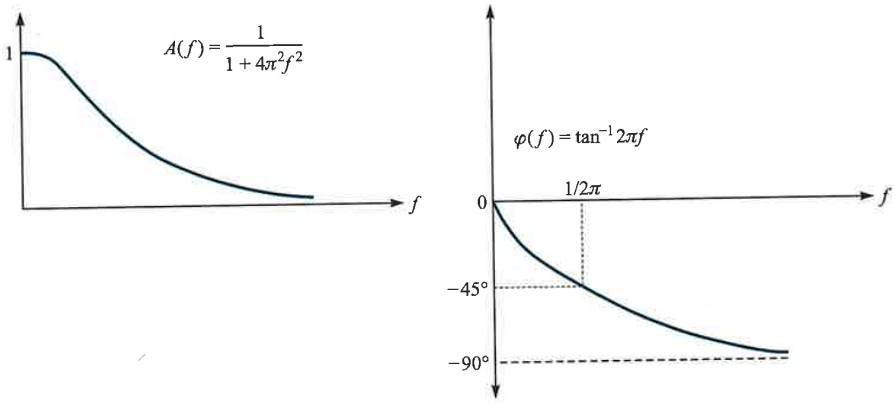


FIGURE 3.14 Example of amplitude-response function and phase-shift function

The **attenuation** of a signal is defined as the reduction or loss in signal power as it is transferred across a system. The attenuation is usually expressed in dB:

$$\text{attenuation} = 10\log_{10}\frac{P_{in}}{P_{out}}$$

The power in a sinusoidal signal of amplitude is $A^2/2$. Therefore, the attenuation in the channel in Figure 3.13 is given by $P_{in}/P_{out} = A_{in}^2/A_{out}^2 = 1/A^2(f)$.

The amplitude-response function $A(f)$ can be viewed as specifying a window of frequencies that the channel will pass. The **bandwidth** W of a channel measures the width of the window of frequencies that are passed by the channel. Figure 3.11 shows two typical amplitude-response functions. The low-pass channel passes low-frequency components and increasingly attenuates higher frequency components. In principle this channel has infinite bandwidth, since every frequency is passed to some degree. However, for practical purposes, signals above the specified frequency are considered negligible. We can define such a frequency as the bandwidth and approximate the amplitude-response function by the idealized low-pass function in Figure 3.11a. Another typical amplitude-response function is a "band pass" channel that passes frequencies in the range f_1 to f_2 instead of low frequencies (see Figure 3.27). The bandwidth for such a channel is $W = f_2 - f_1$.

Communication systems make use of electronic circuits to modify the frequency components of an input signal. When circuits are used in this manner, they are called *filters*. Communication systems usually involve a tandem arrangement of a transmitter filter, a communication channel, and a receiver filter. The overall tandem arrangement can be represented by an overall amplitude-response function $A(f)$ and a phase-shift function $\varphi(f)$. The transmitter and receiver filters are designed to give the overall system the desired amplitude-response and delay properties. For example, devices called loading coils were

added to telephone wire pairs to provide a flat amplitude-response function in the frequency range where telephone voice signals occur. Unfortunately, these coils also introduced a much higher attenuation at the higher frequencies, greatly reducing the bandwidth of the overall system.

Let us now consider the impact of communication channels on other signals. The effect of a channel on an arbitrary input signal can also be determined from $A(f)$ and $\varphi(f)$ as follows. Many signals can be represented as the sum of sinusoidal signals:

$$x(t) = \sum a_k \cos(2\pi f_k t).$$

For example, periodic functions have the preceding form with $f_k = kf_0$ where f_0 is the fundamental frequency.

Now suppose that a periodic signal $x(t)$ is applied to a channel with a channel characterized by $A(f)$ and $\varphi(f)$. The channel attenuates the sinusoidal component at frequency kf_0 by $A(kf_0)$, and it also phase shifts the component by $\varphi(kf_0)$. The output signal of the channel, which we assume to be linear, will therefore be

$$y(t) = \sum a_k A(kf_0) \cos(2\pi k f_0 t + \varphi(kf_0)).$$

This expression shows how the channel distorts the input signal. In general, the amplitude-response function varies with frequency, and so the channel alters the relative weighting of the frequency components. In addition, the different frequency components will be delayed by different amounts, altering the relative alignment between the components. Not surprisingly, then, the shape of the output $y(t)$ generally differs from $x(t)$. Note that the output signal will have its frequencies restricted to the range where the amplitude-response function is nonzero. Thus the bandwidth of the output signal is necessarily less than that of the channel. Note also that if $A(f)$ is equal to a constant, say, C, and if $\varphi(f) = 2\pi f t_d$, over the range of frequencies where a signal $x(t)$ has nonnegligible components, then the output $y(t)$ will be equal to the input signal scaled by the factor C and delayed by t_d seconds.

Example—Effect of a channel on the shape of the output signal

Suppose that binary information is transmitted at a rate of 8 kilobits/second. A binary 1 is transmitted by sending a rectangular pulse of amplitude 1 and of duration 0.125 milliseconds, and a 0 by sending a pulse of amplitude -1. Consider the signal $x_3(t)$ in Figure 3.15 that corresponds to the repetition of the pattern 10000001 over and over again. This periodic signal can be expressed as a sum of sinusoids as follows using Fourier series:

$$x_3(t) = -0.5 + \left(\frac{4}{\pi}\right)\left\{\sin\left(\frac{\pi}{4}\right)\cos(2\pi 1000t) + \sin\left(\frac{2\pi}{4}\right)\cos(2\pi 2000t) + \sin\left(\frac{3\pi}{4}\right)\cos(2\pi 3000t) + \ldots\right\}$$

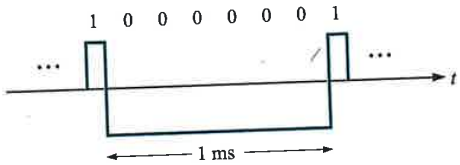


FIGURE 3.15 Signals corresponding to repeated octet patterns

Suppose that the signal is passed through a communication channel that has $A(f) = 1$ and $\varphi(f) = 0$ for f in the range 0 to W. Figures 3.16a, b, and c show the output (the solid line) of the communication channel for values of W (1.5 kHz, 2.5 kHz, and 4.5 kHz) that pass the frequencies only to the first, second, and fourth harmonic, respectively. As the bandwidth of the channel increases, more of the harmonics are passed and the output of the channel more closely approximates the input. This example shows how the bandwidth of the channel affects the ability to transmit digital information in the form of pulses. Clearly, *as bandwidth is decreased, the precision with which the pulses can be identified is reduced.*

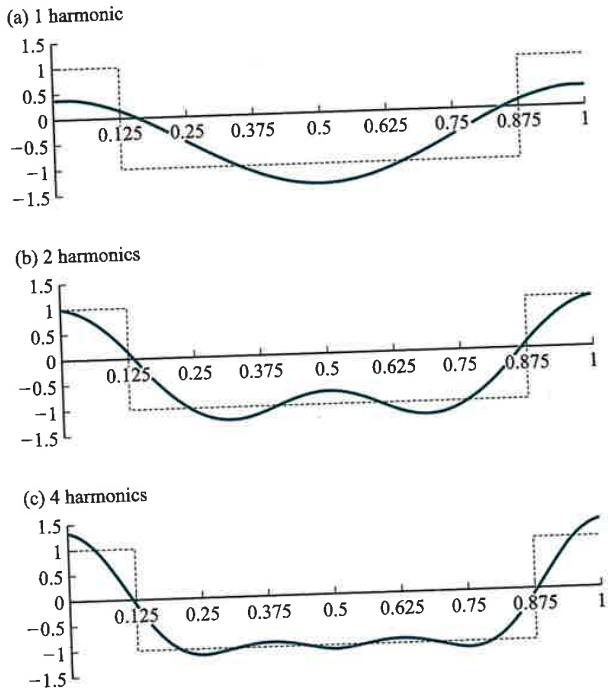


FIGURE 3.16 Output of low-pass communication channels for input signal in Figure 3.15

3.3.2 Time Domain Characterization

Figure 3.17 considers the time domain characterization of a communication channel. A very narrow pulse is applied to the channel at time $t = 0$. The energy associated with the pulse appears at the output of the channel as a signal $h(t)$ some propagation time later. The propagation speed, of course, cannot exceed the speed of light in the given medium. The signal $h(t)$ is called the **impulse response** of the channel. Invariably the output pulse $h(t)$ is spread out in time. The width of the pulse is an indicator of how quickly the output follows the input and hence of how fast pulses can be transmitted over the channel. In digital transmission we are interested in maximizing the number of pulses transmitted per second in order to maximize the rate at which information can be transmitted. Equivalently, we are interested in minimizing the time T between consecutive input pulses. This minimum spacing is determined by the degree of interference between pulses at the output of the channel.

Suppose that we place transmitter and receiver filters around a communication channel and suppose as well that we are interested in using the frequencies in the range 0 to W Hz. Furthermore, suppose that the filters can be selected so that the overall system is an idealized low-pass channel; that is, $A(f) = 1$, and $\varphi(f) = 2\pi f t_d$. It can be shown that the impulse response of the system is given by

$$h(t) = s(t - t_d)$$

which is a delayed version of

$$s(t) = \frac{\sin(2\pi W t)}{2\pi W t}$$

Figure 3.18 shows $s(t)$. It can be seen that this function is equal to 1 at $t = 0$ and that it has zero crossings at nonzero integer multiples of $T = 1/2W$. Note that the pulse is mostly confined to the interval from $-T$ to T, so it is approximately $2T = 2/2W = 1/W$ seconds wide. Thus we see that *as the bandwidth* W *increases, the width of the pulse* s(t) *decreases, suggesting that pulses can be input into the system more closely spaced, that is, at a higher rate.* The next section shows that the signal $s(t)$ plays an important role in the design of digital transmission systems.

Recall that $h(t)$ is the response to a narrow pulse at time $t = 0$, so we see that our ideal system has the strange property that its output $h(t) = s(t - t_d)$ anticipates the input that will be applied and begins appearing at the output before time $t = 0$. In practice, this idealized filter cannot be realized; however, delayed and slightly modified versions of $s(t)$ are approximated and implemented in real systems.

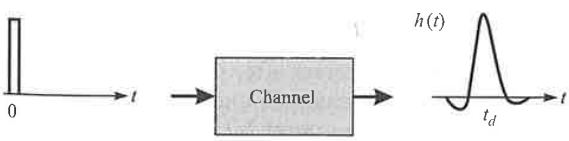


FIGURE 3.17 Channel characterization—time domain

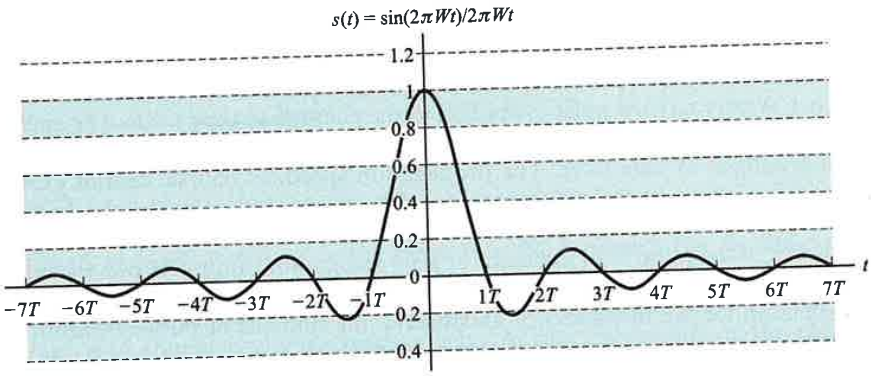


FIGURE 3.18 Signaling pulse with zero intersymbol interference

3.4 FUNDAMENTAL LIMITS IN DIGITAL TRANSMISSION

In this section we consider **baseband transmission**, which is the transmission of digital information over a low-pass communication channel. The quality of a digital transmission system is determined by the transmission rate or bit rate at which information bits can be transmitted reliably. Thus the quality is measured in terms of two parameters: *transmission speed*, or *bit rate*, in bits per second and the *bit error rate*, the fraction of bits that are received in error. We will see that these two parameters are determined by the bandwidth of the communication channel and by the SNR, which we will define formally later in the section.

Figure 3.19 shows the simplest way to transmit a binary information sequence. Every T seconds the transmitter accepts a binary information bit and transmits a pulse with amplitude $+A$ if the information bit is a 1 and with $-A$ if the information bit is a 0. In the examples in section 3.3, we saw how a channel distorts an input signal and limits the ability to correctly detect the polarity of a pulse. In this section we show how the problem of channel distortion is addressed and how the pulse transmission rate is maximized at the same time. In particular, in Figure 3.19 each pulse at the input results in a pulse at the output. We also show how the pulses that arrive at the receiver can be packed as closely as possible if they are shaped appropriately by the transmitter and receiver filters.

3.4.1 The Nyquist Signaling Rate

Let $p(t)$ be the basic pulse that appears at the receiver after it has been sent over the combined transmitter filter, communication channel, and receiver filter. The first pulse is transmitted, centered at $t = 0$. If the input bit was 1, then $+Ap(t)$

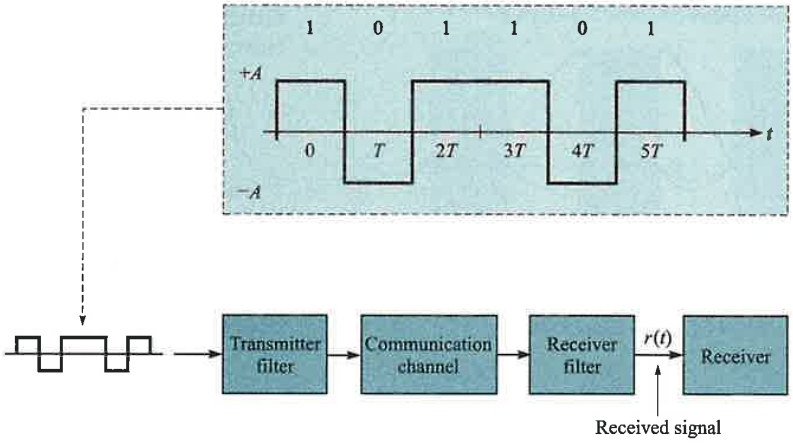


FIGURE 3.19 Digital baseband signal and baseband transmission system

should be received; if the input was 0, then $-Ap(t)$ should be received instead. For simplicity, we assume that the propagation delay is zero. To determine what was sent at the transmitter, the receiver samples the signal it receives at $t = 0$. If the sample is positive, the receiver decides that a 1 was sent; if the sample is negative, the receiver decides that a 0 was sent.

Every T seconds the transmitter sends an additional information bit by transmitting another pulse with the appropriate polarity. For example, the second bit is sent at time $t = T$ and will be either $+Ap(t - T)$ or $-Ap(t - T)$, depending on the information bit. The receiver samples its signal at $t = T$ to determine the corresponding input. However, the pulses are sent as part of a sequence, and so the total signal $r(t)$ that appears at the receiver is the *sum* of all the inputs:

$$r(t) = \sum_k A_k p(t - kT)$$

According to this expression, when the receiver samples the signal at $t = 0$, it measures

$$r(0) = A_0 p(0) + \sum_{k \neq 0} A_k p(-kT)$$

In other words, the receiver must contend with *intersymbol interference* from all the other transmitted pulses. What a mess! Note, however, that all the terms in the summation disappear if we use a pulse that has zero crossings at $t = kT$ for nonzero integer values k. The pulse $s(t)$ introduced in section 3.3.2 and shown in Figure 3.18 satisfies this property. This pulse is an example of the class of *Nyquist pulses* that have the property of providing *zero intersymbol interference* at the times $t = kT$ at the receiver. Figure 3.20a shows the three pulses

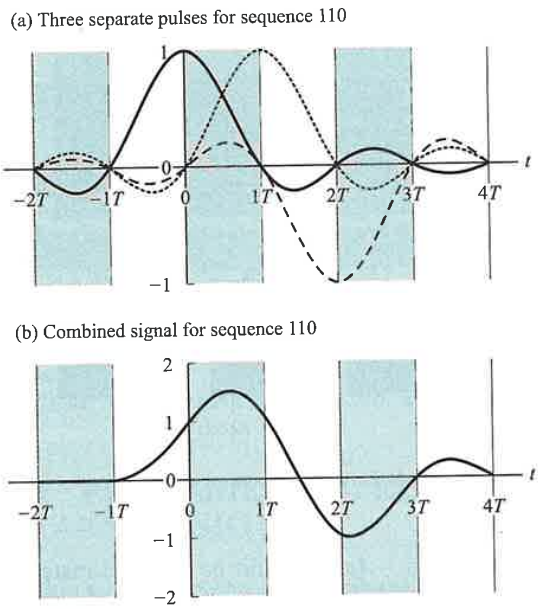


FIGURE 3.20 System response to binary input 110

corresponding to the sequence 110, before they are added. Figure 3.20b shows the signal that results when the three pulses are combined. It can be seen that the combined signal has the correct values at $t = 0$, 1, and 2.

The above transmission system sends a bit every T seconds, where $T = 1/2W$ and W is the bandwidth of the overall system in Figure 3.19. For example, if $W = 4$ kHz, then pulses would be sent every $T = 1/8000 = 125$ microseconds, which corresponds to a rate of 8000 pulses/second. A bit is sent with every pulse, so the bit rate is 8000 bits/second. The **Nyquist Signaling Rate** is defined by

$$r_{max} = 2W \text{pulses/second}$$

The Nyquist rate r_{max} is the maximum signaling rate that is achievable through an ideal low-pass channel with no intersymbol interference.

We already noted that the ideal low-pass system in Figure 3.11 cannot be implemented in practice. Nyquist also found other pulses that have zero intersymbol interference but that require some additional bandwidth. Figure 3.21 shows the amplitude-response function for one such pulse. Here a transition region with odd symmetry about $f = W$ is introduced. The more gradual roll-off of these systems makes the appropriate transmitter and receiver filters simpler to attain in practice.

The operation of the baseband transmission systems in this section depends critically on having the receiver synchronized precisely to intervals of duration T. Additional processing of the received signal is carried by the receiver to obtain this synchronization. If the receiver loses synchronization, then it will start sampling the signal at time instants that do contain intersymbol interference. The use

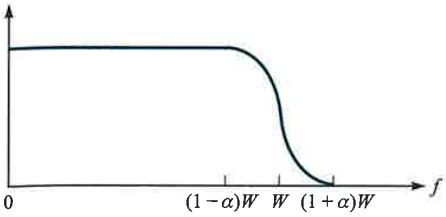


FIGURE 3.21 Raised cosine transfer function

of pulses corresponding to the system in Figure 3.21 provides some tolerance to small errors in sampling time.

3.4.2 The Shannon Channel Capacity

Up to this point we have been assuming that the input pulses can have only two values, 0 or 1. This restriction can be relaxed, and the pulses can be allowed to assume a greater number of values. Consider **multilevel transmission** where binary information is transmitted in a system that uses one of 2^m distinct levels in each input pulse. The binary information sequence can be broken into groups of m bits. Proceeding as before, each T seconds the transmitter accepts a group of m bits. These m bits determine a unique amplitude of the pulse that is to be input into the system. As long as the signaling rate does not exceed the Nyquist rate $2W$, the interference between pulses will still be zero, and by measuring the output at the right time instant we will be able to determine the input. Thus if we suppose that we transmit $2W$ pulses/second over a channel that has bandwidth W and if the number of amplitude values is 2^m, then the **bit rate** R of the system is

$$R = 2W \text{ pulses/second} * m \text{ bits/pulse} = 2Wm \text{ bits/second}$$

In principle we can attain arbitrarily high bit rates by increasing the number of levels 2^m. However, we cannot do so in practice because of the limitations on the accuracy with which measurements can be made and also the presence of random noise. The random noise implies that the value of the overall response at time $t = kT$ will be the sum of the input amplitude plus some random noise. This noise can cause the measurement system to make an incorrect decision. *To keep the probability of decision errors small, we must maintain some minimum spacing between amplitude values* as shown in Figure 3.22. Here four signal levels are shown next to the typical noise. In the case of four levels, the noise is not likely to cause errors. Figure 3.22 also shows a case with eight signal levels. It can be seen that if the spacing between levels becomes too small, then the noise signals can cause the receiver to make the wrong decision.

We can make the discussion more precise by considering the statistics of the noise signal. Figure 3.23 shows the Gaussian probability density function, which is frequently a good model for the noise amplitudes. The density function gives

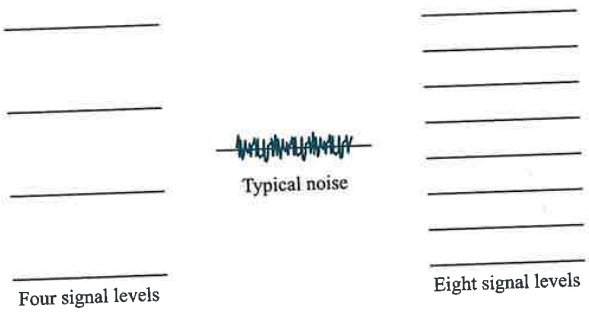


FIGURE 3.22 Effect of noise on transmission errors as number of levels is increased

the relative frequency of occurrence of the noise amplitudes. It can be seen that for the Gaussian density, the amplitudes are centered around zero. The average power of this noise signal is given by σ^2, where σ is the standard deviation.

Consider how errors occur in multilevel transmission. If we have maximum amplitudes $\pm A$ and M levels, then the separation between adjacent levels is $\delta = 2A/(M-1)$. When an interior signal level is transmitted, an error occurs if the noise causes the received signal to be closer to one of the other signal levels. This situation occurs if the noise amplitude is greater than $\delta/2$ or less than $-\delta/2$. Thus the probability of error for an interior signal level is given by

$$P_e = \int_{-\infty}^{-\delta/2} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} dx + \int_{\delta/2}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} dx = 2 \int_{\delta/2\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

$$= 2Q\left(\frac{\delta}{2\sigma}\right)$$

The expression on the right-hand side is evaluated using tables or analytic aproximations [Leon-Garcia, 1994, chapter 3]. Figure 3.24 shows how P_e varies with $\delta/2\sigma = A/(M-1)\sigma$. For larger values of separation $\delta/2\sigma$, large decreases in the probability of error are possible with small increases in $\delta/2\sigma$. However, as the number of signal levels M is increased, $\delta/2\sigma$ is decreased, leading to large increases in the probability of error. We conclude that the bit rate cannot be increased to arbitrarily high values by increasing M without incurring significantly higher bit error rates.

We have now seen that two parameters affect the performance of a digital transmission system: bandwidth and SNR. Shannon addressed the question of determining the maximum achievable bit rate at which reliable communication is

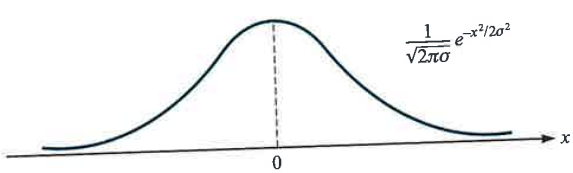


FIGURE 3.23 Gaussian probability density function

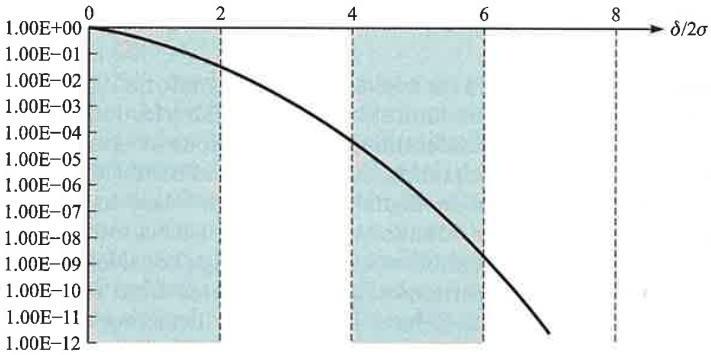


FIGURE 3.24 Probability of error for an interior signal level

possible over an ideal channel of bandwidth W and of a given SNR. The phrase *reliable communication* means that it is possible to achieve arbitrarily small error probabilities by using sufficiently complex coding. Shannon derived the channel capacity for such a channel under the condition that the noise has a Gaussian distribution. This **channel capacity** is given by the following formula:

$$C = W \log_2(1 + \text{SNR}) \text{ bits/second}$$

Shannon showed that the probability of error can be made arbitrarily small only if the transmission rate R is less than channel capacity C. Therefore, the channel capacity is the maximum possible transmission rate over a system with given parameters.

SHANNON CHANNEL CAPACITY OF TELEPHONE CHANNEL

Consider a telephone channel with W = 3.4 kHz and SNR = 10,000. The channel capacity is then

$$C = 3400 \log_2(1 + 10000) = 44{,}800 \text{ bits/second}$$

The following identities are useful here: $\log_2 x = \ln x/\ln 2 = \log_{10} x/\log_{10} 2$. We note that the SNR is usually stated in dB. Thus if SNR = 10,000, then in dB the SNR is

$$10 \log_{10} \text{SNR dB} = 10 \log_{10} 10000 = 40 \text{ dB}$$

The above result gives a bound to the achievable bit rate over ordinary analog telephone lines.

3.5 LINE CODING

Line coding is the method used for converting a binary information sequence into a digital signal in a digital communications system. The selection of a line coding technique involves several considerations. In the previous sections, we focused on maximizing the bit rate over channels that have limited bandwidths. Maximizing bit rate is the main concern in digital transmission when bandwidth is at a premium. However, in other situations, such as in LANs, other concerns are also of interest. For example, another important design consideration is the ease with which the bit timing information can be recovered from the digital signal. Also, some line coding methods have built-in error detecting capabilities, and other methods have better immunity to noise and interference. Finally, the complexity and the cost of the line code implementations are always factors in the selection for a given application.

Figure 3.25 shows various line codes that are used in practice. The figure shows the digital signals that are produced by the line codes for the binary sequence 101011100. The simplest scheme is the unipolar **nonreturn-to-zero (NRZ) encoding** in which a binary 1 is transmitted by sending a $+A$ voltage level, and a 0 is transmitted by sending a 0 voltage. If binary 0s and 1s both occur with probability 1/2, then the average transmitted power for this line code is $(1/2)A^2 + (1/2)0^2 = A^2/2$. The **polar NRZ encoding** method that maps a binary 1 to $+A/2$ and binary 0 to $-A/2$ is more efficient than unipolar NRZ in

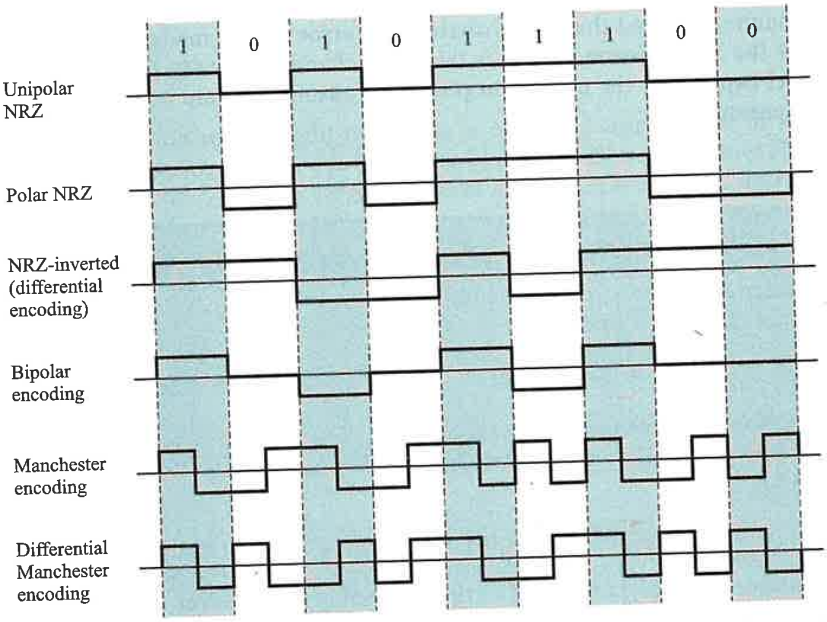


FIGURE 3.25 Line coding methods

terms of average transmitted power. Its average power is given by $(1/2)(+A/2)^2 + (1/2)(-A/2)^2 = A^2/4$.

The spectrum that results from applying a given line code is of interest. We usually assume that the binary information is equally likely to be 0 or 1 and that they are statistically independent of each other, much as if they were produced by a sequence of independent coin flips. The unipolar and the polar NRZ encoding methods have the same frequency components because they produce essentially the same variations in a signal as a function of time. Strings of consecutive 0s and consecutive 1s lead to periods where the signal remains constant. These strings of 0s and 1s occur frequently enough to produce a spectrum that has its components concentrated at the lower frequencies as shown in Figure 3.26.[7] This situation presents a problem when the communications channel does not pass low frequencies. For example, most telephone transmission systems do not pass the frequencies below about 200 Hz.

The **bipolar encoding** method was developed to produce a spectrum that is more amenable to channels that do not pass low frequencies. In this method binary 0s are mapped into 0 voltage, thus making no contribution to the digital signals; consecutive 1s are alternately mapped into $+A/2$ and $-A/2$. Thus a string of consecutive 1s will produce a square wave with the frequency $1/2T$ Hz. As a result, the spectrum for the bipolar code has its frequency content centered around the frequency $1/2T$ Hz and has small content at low frequencies as shown in Figure 3.26.

Timing recovery is an important consideration in the selection of a line code. The timing-recovery circuit in the receiver monitors the transitions at the edge of the bit intervals to determine the boundary between bits. Long strings of 0s and

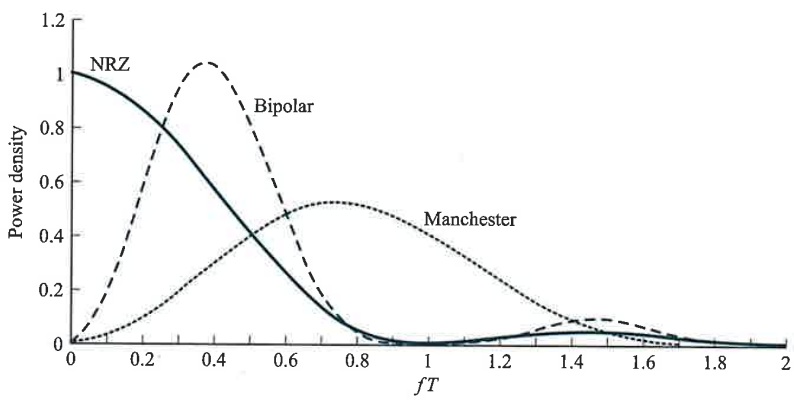


FIGURE 3.26 Spectra for different line codes

[7]The formulas for the spectra produced by the line codes in Figure 3.26 can be found in [Smith 1985, pp. 198–203].

1s in the binary and the polar binary encodings can cause the timing circuit to lose synchronization because of the absence of transitions. In the bipolar encoding long strings of 1s result in a square wave that has strong timing content; however, long strings of 0s still pose a problem. To address this problem, the bipolar line codes used in telephone transmission systems place a limit on the maximum number of 0s that may be encoded into the digital signal. Whenever a string of N consecutive 0s occurs, the string is encoded into a special binary sequence that contains 0s and 1s. To alert the receiver that a substitution has been made, the sequence is encoded so that the mapping in the bipolar line code is violated; that is, two consecutive 1s do not alternate in polarity.

A problem with polar coding is that a systematic error in polarity can cause all 0s to be detected as 1s and all 1s as 0s.[8] The problem can be avoided by mapping the binary information into *transitions* at the beginning of each interval. A binary 1 is transmitted by enforcing a transition at the beginning of a bit time, and a 0 by having no transition. The signal level within the actual bit time remains constant. Figure 3.25 shows an example of how **differential encoding**, or **NRZ inverted**, carries out this mapping. Starting at a given level, the sequence of bits determines the subsequent transitions at the beginning of each interval. Note that differential encoding will lead to the same spectrum as binary and polar encoding. However, errors in differential encoding tend to occur in pairs. An error in one bit time will provide the wrong reference for the next time, thus leading to an additional error in the next bit.

Example—Ethernet and Token-Ring Line Coding

Bipolar coding has been used in long-distance transmission where bandwidth efficiency is important. In LANs, where the distances are short, bandwidth efficiency is much less important than cost per station. The Manchester encodings shown in Figure 3.25 are used in Ethernet and token-ring LAN standards. In **Manchester encoding** a binary 1 is denoted by a transition from $A/2$ to $-A/2$ in the middle of the bit time interval, and a binary 0 by a transition from $-A/2$ to $A/2$. The presence of a transition in the middle of every bit interval makes timing recovery particularly easy and also results in small content at low frequencies. However, the pulse rate is essentially double that of binary encoding, and this factor results in a spectrum with significantly larger bandwidth as shown in Figure 3.26. **Differential Manchester encoding**, which is used in token-ring networks, retains the transition in the middle of every bit time, but the binary sequence is mapped into the presence or absence of transitions in the beginning of the bit intervals. In this type of encoding, a binary 0 is marked by a transition at the beginning of an interval, whereas a 1 is marked by the absence of a transition.

---

[8]This polarity inversion occurs when the polar-encoded stream is fed into a phase modulation system such as the one discussed in section 3.6.

Note that the Manchester encoding can be viewed as the transmission of two pulses for each binary bit. A binary 1 is mapped into the binary pair of 10, and the corresponding polar encoding for these two bits is transmitted: A binary 0 is mapped into 01. The Manchester code is an example of a mBnB code (where m is 1 and n is 2) in which m information bits are mapped into $n > m$ encoded bits. The encoded bits are selected so that they provide enough pulses for timing recovery and limit the number of pulses of the same level. For example, the optical fiber transmission system in the Fiber Distributed Data Interface (FDDI) LAN uses a 4B5B line code.

3.6 MODEMS AND DIGITAL MODULATION

In section 3.4 we considered digital transmission over channels that are low pass in nature. We now consider band-pass channels that do not pass the lower frequencies and instead pass power in some frequency range from f_1 to f_2, as shown in Figure 3.27. We assume that the bandwidth of the channel is $W = f_2 - f_1$ and discuss the use of modulation to transmit digital information over this type of channel. The basic function of the modulation is to produce a signal that contains the information sequence and that occupies frequencies in the range passed by the channel. A **modem** is a device that carries out this basic function. In this section we first consider the principles of digital modulation, and then we show how these principles are applied in telephone modem standards.

Let f_c be the frequency in the center of the band-pass channel in Figure 3.27; that is, $f_c = (f_1 + f_2)/2$. The sinusoidal signal $\cos(2\pi f_c t)$ has all of its power located precisely at frequency f_c. The various types of modulation schemes involve imbedding the binary information sequence into the transmitted signal by varying, or modulating, some attribute of the sinusoidal signal. In **amplitude shift keying (ASK)** the sinusoidal signal is turned on and off according to the information sequence as shown in Figure 3.28a. The demodulator for an ASK system needs only to determine the presence or absence of a sinusoid in a given time interval. In **frequency shift keying (FSK)**, shown in Figure 3.28b, the frequency of the sinusoid is varied according to the information. If the information bit is a 0, the sinusoid has frequency $f_1 = f_c - \varepsilon$, and if it is a 1, the sinusoid has a frequency $f_2 = f_c + \varepsilon$. The demodulator for an FSK system must be able to determine which of two possible frequencies is present at a given time. In

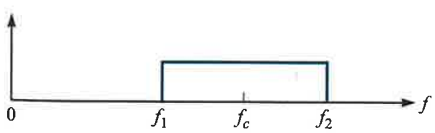


FIGURE 3.27 Digital modulation

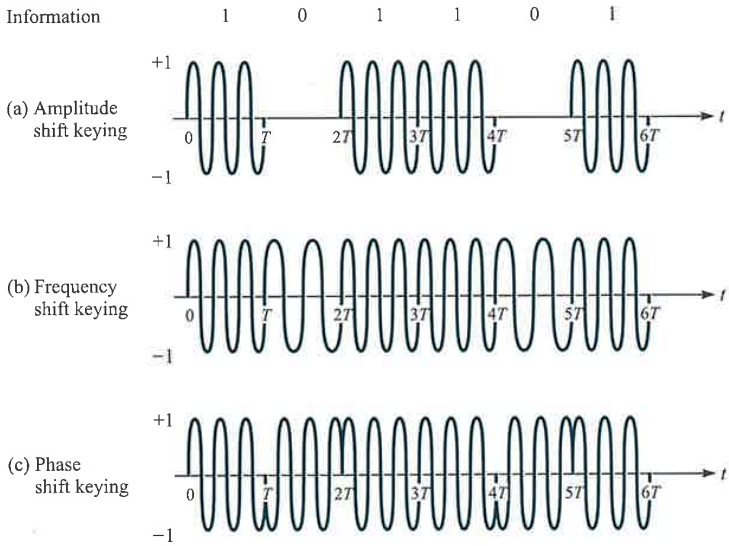


FIGURE 3.28 Amplitude, frequency, and phase modulation techniques

phase shift keying (PSK), the phase of the sinusoid is altered according to the information sequence. In Figure 3.28c a binary 1 is transmitted by $\cos(2\pi f_c t)$, and a binary 0 is transmitted by $\cos(2\pi f_c t + \pi)$. Because $\cos(2\pi f_c t + \pi) = -\cos(2\pi f_c t)$, we note that this PSK scheme is equivalent to multiplying the sinusoidal signal by $+1$ when the information is a 1 and by -1 when the information bit is a 0. Thus the demodulator for a PSK system must be able to determine the phase of the received sinusoid with respect to some reference phase. In the remainder of this section, we concentrate on phase modulation techniques.

Consider the problem of transmitting a binary information sequence over an ideal band-pass channel using PSK. We are interested in developing modulation techniques that can achieve pulse rates that are comparable to that achieved by Nyquist signaling over low-pass channels. In Figure 3.29c we show the waveform $Y_i(t)$ that results when a binary 1 is transmitted using a cosine wave with amplitude $+A$, and a binary 0 is transmitted using a cosine wave with amplitude $-A$. The corresponding modulator is shown in Figure 3.30a. Every T seconds the modulator accepts a new binary information symbol and adjusts the amplitude A_k accordingly. In effect, as shown in Figure 3.29c, the modulator transmits a T-second segment of the signal as follows:

$+A\cos(2\pi f_c t)$ if the information symbol is a 1.

$-A\cos(2\pi f_c t)$ if the information symbol is a 0.

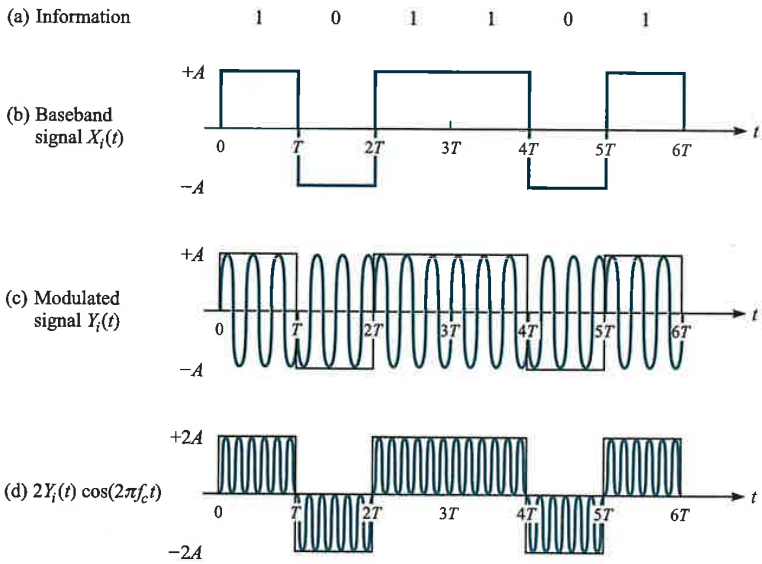


FIGURE 3.29 Modulating a signal

Note that the modulated signal is no longer a pure sinusoid, since the overall transmitted signal contains glitches between the T-second intervals, but its primary oscillations are still around the center frequency f_c; therefore, we expect that the power of the signal wil be centered about f_c and hence located in the range of frequencies that are passed by the band-pass channel.

By monitoring the polarity of the signal over the intervals of T seconds, a receiver can recover the original information sequence. Let us see more precisely how this recovery may be accomplished. As shown in Figure 3.30b suppose we

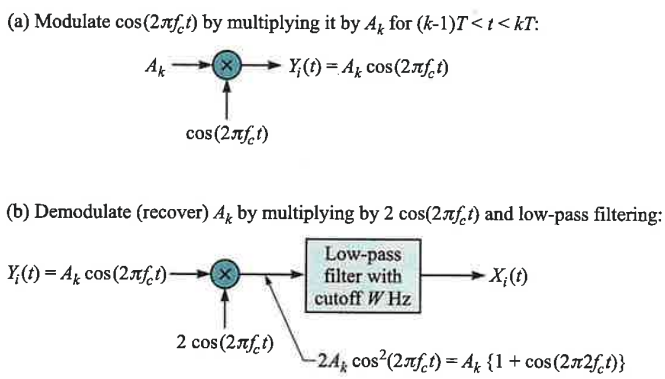


FIGURE 3.30 Modulator and demodulator

multiply the modulated signal $Y_i(t)$ by $2\cos(2\pi f_c t)$. The resulting signal is $+2A\cos^2(2\pi f_c t)$ if the original information symbol is a 1 or $-2A\cos^2(2\pi f_c t)$ if the original information symbol is 0. Because $2\cos^2(2\pi f_c t) = (1 + \cos(4\pi f_c t))$, we see that the resulting signals are as shown in Figure 3.29d. By smoothing out the oscillatory part with a so-called low-pass filter, we can easily determine the original baseband signal $X_i(t)$ and the A_k and subsequently the original binary sequence.

When we developed the Nyquist signal result in section 3.4.1, we found that for a low-pass channel of bandwidth W Hz the maximum signaling rate is $2W$ pulses/second. It can be shown that the system we have just described can transmit only W pulses/seconds over a band-pass channel that has bandwidth W.[9] Consequently, the time per bit is given by $T = 1/W$. Thus this scheme attains only half the signaling rate of the low-pass case. Next we show how we can recover this factor of 2 by using Quadrature Amplitude Modulation.

Suppose we have an original information stream that is generating symbols at a rate of $2W$ symbols per second. In **Quadrature Amplitude Modulation (QAM)** we split the original information stream into two sequences that consist of the odd and even symbols, say, B_k and A_k, respectively, as shown in Figure 3.31. Each sequence now has the rate W symbols per second. Suppose we take the even sequence A_k and produce a modulated signal by multiplying it by $\cos(2\pi f_c t)$; that is, $Y_i(t) = A_k \cos(2\pi f_c t)$ for a T-second interval. As before, this modulated signal will be located within the band of the band-pass channel. Now suppose that we take the odd sequence B_k and produce another modulated signal by multiplying it by $\sin(2\pi f_c t)$; that is, $Y_q(t) = B_k \sin(2\pi f_c t)$ for a T-second interval. This modulated signal also has its power located within the band of the band-pass channel. We finally obtain a composite modulated signal by adding $Y_i(t)$ and $Y_q(t)$, as shown in Figure 3.31.

$$Y(t) = Y_i(t) + Y_q(t) = A_k \cos(2\pi f_c t) + B_k \sin(2\pi f_c t).$$

This equation shows that we have generated what amounts to a two-dimensional modulation scheme. The first component A_k is called the **in-phase component**; the second component B_k is called the **quadrature-phase component**.

We now transmit the sum of these two modulated signals over the band-pass channel. The composite sinusoidal signal $Y(t)$ will be passed without distortion by the linear band-pass channel. We now need to demonstrate how the original information symbols can be recovered from $Y(t)$. We will see that our ability to do so depends on the following properties of cosines and sines:

$$2\cos^2(2\pi f_c t) = 1 + \cos(4\pi f_c t)$$
$$2\sin^2(2\pi f_c t) = 1 - \sin(4\pi f_c t)$$
$$2\cos(2\pi f_c t)\sin(2\pi f_c t) = 0 + \sin(4\pi f_c t)$$

[9] In the remainder of this section, we continue to use the term *pulse* for the signal that is transmitted in a T-second interval.

Ex. 1015.57

Modulate $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ by multiplying them by A_k and B_k, respectively, for $(k\text{-}1)T < t < kT$:

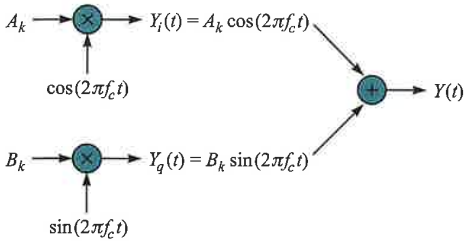


FIGURE 3.31 QAM modulator

These properties allow us to recover the original symbols as shown in Figure 3.32. By multiplying $Y(t)$ by $2\cos(2\pi f_c t)$ and then low-pass filtering the resulting signal, we obtain the sequence A_k. Note that the cross-product term $B_k(t)\sin(4\pi f_c t)$ is removed by the low-pass filter. Similarly, the sequence B_k is recovered by multiplying $Y(t)$ by $2\sin(2\pi f_c t)$ and low-pass filtering the output. Thus QAM is a two-dimensional system that achieves an effective signaling rate of $2W$ pulses per second over the band-pass channel of W Hz. This result matches the performance of the Nyquist signaling procedure that we developed in section 3.4.

The two-dimensional nature of the above signaling scheme can be used to plot the various combinations of levels that are allowed in a given signaling interval of T seconds. (Note: It is important to keep in mind that $T = 1/W$ in this discussion). In the case considered so far, the term multiplying the cosine function can assume the value $+A$ or $-A$; the term multiplying the sine function can also assume that value $+A$ or $-A$. In total, four combinations of these values can occur. These are shown as the four points in the two-dimensional plane in Figure 3.33. At any given T-second interval, only one of the four points in this **signal constellation** can be in use. It is therefore clear that in every T-second interval we are transmitting two bits of information. As in the case of baseband

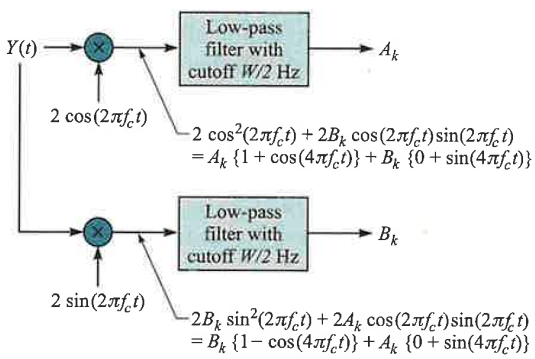


FIGURE 3.32 QAM demodulator

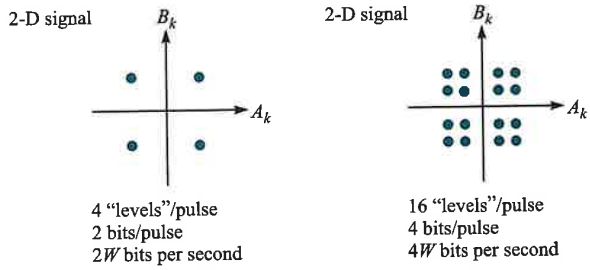


FIGURE 3.33 Signal constellations

signaling, we can increase the number of bits that can be transmitted per T-second interval by increasing the number of levels that are used. Figure 3.33 shows a 16-point constellation that results when the terms multiplying the cosine and sine functions are allowed to assume four possible levels. In this case only one of the 16 points in the constellation is in use in any given T-second interval, and hence four bits of information are transmitted at every such interval.

Another way of viewing QAM is as the simultaneous modulation of the amplitude and phase of a carrier signal, since

$$A_k \cos(2\pi f_c t) + B_k \sin(2\pi f_c t) = (A_k^2 + B_k^2)^{1/2} \cos(2\pi f_c t + \tan^{-1} B_k/A_k)$$

Each signal constellation point can then be seen as determining a specific amplitude and phase.

Many signal constellations that are used in practice have nonrectangular arrays of signal points such as those shown in Figure 3.34. To produce this type of signaling, we need to modify the above encoding scheme only slightly. Suppose that the constellation has 2^m points. Each T-second interval, the transmitter accepts m information bits, identifies the constellation point assigned to these bits, and then transmits cosine and sine signals with the amplitudes that correspond to the constellation point.

The presence of noise in transmission systems implies that the pair of recovered values for the cosine and sine components will differ somewhat from the transmitted values. This pair of values will therefore specify a point in the plane

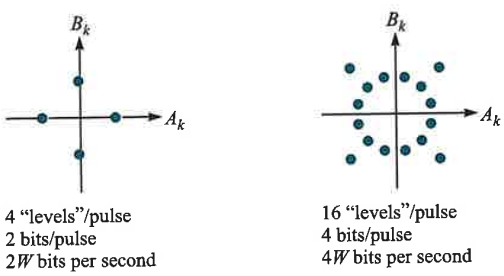


FIGURE 3.34 Other signal constellations

that deviates from the transmitted constellation point. The task of the receiver is to take the received pair of values and identify the closest constellation point.

3.6.1 Signal Constellations and Telephone Modem Standards

Signal constellation diagrams of the type shown in Figure 3.33 and Figure 3.34 are used in the various signaling standards that have been adopted for use over telephone lines. For the purposes of data communications, most telephone channels have a usable bandwidth in the range $f_1 = 500$ Hz to $f_2 = 2900$ Hz. This implies $W = 2400$ and hence a signaling rate of $1/T = W = 2400$ pulses/second. Table 3.4 lists some of the parameters that specify the ITU V.32bis and V.34bis modem standards that are in current use. Each standard can operate at a number of speeds that depend on the quality of the channel available. Both standards operate at a rate of 2400 pulses/second, and the actual bit rate is determined by which constellation is used. The QAM 4 systems uses four constellation points and hence two bits/pulse, giving a bit rate of 4800 bps.

Trellis modulation systems are more complex in that they combine error-correction coding with the modulation. In trellis modulation the number of constellation points is 2^{m+1}. At every T-second interval, the trellis coding algorithm accepts m bits and generates $m + 1$ bits that specify the constellation point that is to be used. In effect, only 2^m out of the 2^{m+1} possible constellation points are valid during any given interval. This extra degree of redundancy improves the robustness of the modulation scheme with respect to errors. In Table 3.4, the trellis 32 system has 2^5 constellation points out of which 16 are valid at any given time; thus the bit rate is $4 \times 2400 = 9600$ bps. Similarly, the trellis 128 system gives a bit rate of $6 \times 2400 = 14{,}400$ bps.

The V.34bis standard can operate at rates of 2400, 2743, 2800, 3000, 3200, or 3429 pulses/second. The modem precedes communications with an initial phase during which the channel is probed to determine the usable bandwidth in the given telephone connection. The modem then selects a pulse rate. For each of these pulse rates, a number of possible trellis encoding schemes are defined. Each encoding scheme selects a constellation that consists of a subset of points from a superconstellation of 860 points. A range of bit rates are possible, including 2400, 4800, 9600, 14,400, 19,200 and 28,800, 31,200 and 33,600 bps.

V.32bis	Modulation	Pulse rate
14,000 bps	Trellis 128	2400 pulses/second
9600 bps	Trellis 32	2400 pulses/second
4800 bps	QAM 4	2400 pulses/second
V.34bis		
2400–33,600 bps	Trellis 960	2400–3429 pulses/second

TABLE 3.4 Modem standards

It is instructive to consider how close the V.34bis modem comes to the maximum possible transmission rate predicted by Shannon's formula. Suppose we have a maximum useful bandwidth of 3400 Hz and assume a maximum SNR of 40 dB, which is a bit over the maximum possible in a telephone line. Shannon's formula then gives a maximum possible bit rate of 44,880 bits/second. It is clear then that the V.34bis modem is coming close to achieving the Shannon bound. In 1997 a new class of modems, the ITU-T V.90 standard, was introduced that tout a bit rate of 56,000 bits/second, well in excess of the Shannon bound! As indicated at the beginning of this chapter, the 56 kbps speed is attained only under particular conditions that do not correspond to the normal telephone channel.[10]

BEEEEEEP, CHIRP, CHIRP, KTWANG, KTWANG, shhhhh, SHHHHHH

What are the calling and answering V.34 modems up to when they make these noises? They are carrying out the handshaking that is required to set up communication. V.34 handshaking has four phases:

Phase 1: Because the modems don't know anything about each other's characteristics, they begin by communicating using simple, low-speed, 300 bps FSK. They exchange information about the available modulation modes, the standards they support, the type of error correction that is to be used, and whether the connection is cellular.

Phase 2: The modems perform probing of the telephone line by transmitting tones with specified phases and frequencies that are spaced 150 Hz apart and that cover the range from 150 Hz to 3750 Hz. The tones are sent at two distinct signal levels (amplitudes). The probing allows the modems to determine the bandwidth and distortion of the telephone line. The modems then select their carrier frequency and signal level. At the end of this phase, the modems exchange information about their carrier frequency, transmit power, symbol rate, and maximum data rate.

Phase 3: The receiver equalizer starts its adaptation to the channel. The echo canceler is started. The function of the echo canceler is to suppress its modem's transmission signal so that the modem can hear the arriving signal.

Phase 4: The modems exchange information about the specific modem parameters that are to be used, for example, signal constellation, trellis encoding, and other encoding parameters.

The modems are now ready to work for you!

[10]Indeed, when we discuss transmission medium in section 3.7, we show that much higher bit rates are attainable over twisted-wire pairs.

Ex. 1015.61

Modem standards have also been developed for providing error control as well as data compression capabilities. The V.42bis standard specifies the Link Access Procedure for Modems (LAPM) for providing error control. LAPM is based on the HDLC data link control, which is discussed in Chapter 5. V.42bis also specifies the use of the Lempel-Ziv data compression scheme for the compression of information prior to transmission. This scheme can usually provide compression ratios of two or more, thus providing an apparent modem speedup of two or more. The data compression scheme is explained in Chapter 12.

In this section we have considered only telephone modem applications. Digital modulation techniques are also used extensively in other digital transmission systems such as digital cellular telephony and terrestrial and satellite communications. These systems are discussed further in the next section.

3.7 PROPERTIES OF MEDIA AND DIGITAL TRANSMISSION SYSTEMS

For transmission to occur, we must have a transmission medium that conveys the energy of a signal from a sender to a receiver. A communication system places transmitter and receiver equipment on either end of a transmission medium to form a communications channel. In previous sections we discussed how communications channels are characterized in general. In this section we discuss the properties of the transmission media that are used in modern communication networks.

We found that the capability of a channel to carry information reliably is determined by several properties. First, the manner in which the medium transfers signals at various frequencies (that is, the amplitude-response function $A(f)$ and the phase-shift function $\varphi(f)$) determines the extent to which the input pulses are distorted. The transmitter and receiver equipment must be designed to remove enough distortion to make reliable detection of the pulses possible. Second, the transmitted signal is attenuated as it propagates through the medium and noise is also introduced in the medium and in the receiver. These phenomena determine the SNR at the receiver and hence the probability of bit errors. In discussing specific transmission media, we are therefore interested in the following characteristics:

- The amplitude-response function $A(f)$ and the phase-shift function $\varphi(f)$ of the medium and the associated bandwidth as a function of distance.
- The susceptibility of the medium to noise and interference from other sources.

A typical communications system transmits information by modulating a sinusoidal signal of frequency f_0 that is inserted into a guided medium or radiated through an antenna. The sinusoidal variations of the modulated signal propagate in a medium at a speed of v meters/second, where

Ex. 1015.62

$$v = \frac{c}{\sqrt{\varepsilon} f_0}$$

and where $c = 3 \times 10^8$ meters/second is the speed of light in a vacuum and ε is the dielectric constant of the medium. In free space, $\varepsilon = 1$, and $\varepsilon \geq 1$ otherwise. The **wavelength** λ of the signal is given by the length in space spanned by one period of the sinusoid:

$$\lambda = v/f_0 \text{ meters}$$

As shown in Figure 3.35, a 100 MHz carrier signal, which corresponds to FM broadcast radio, has a wavelength of 3 meters, whereas a 3 GHz carrier signal has a wavelength of 10 cm. Infrared light covers the range from 10^{12} to 10^{14} Hz, and the visible light used in optical fiber occupies the range 10^{14} to 10^{15} Hz.

The speed of light c is a maximum limit for propagation speed in free space and cannot be exceeded. Thus if a pulse of energy enters a communications channel of distance d at time $t = 0$, then none of the energy can appear at the output before time $t = d/c$ as shown in Figure 3.36. Note that in copper wire the speed of light is 2.3×10^8 meters/second, and in optical fiber systems the speed of light v is approximately 2×10^8 meters/second.

The two basic types of media are wired media, in which the signal energy is contained and guided within a solid medium, and wireless media, in which the signal energy propagates in the form of unguided electromagnetic signals. Copper pair wires, coaxial cable, and optical fiber are examples of wired media. Radio and infrared light are examples of wireless media.

Wired and wireless media differ in a fundamental way. In its most basic form, wired media provide communications from point to point. By interconnecting wires at various repeater or switching points, wired media lead to well-defined *discrete* network topologies. Since the energy is confined within the medium, additional transmission capacity can be procured by adding more wires. Unguided media, on the other hand, can achieve only limited directionality and can be transmitted, as in the case of broadcast radio, in all directions

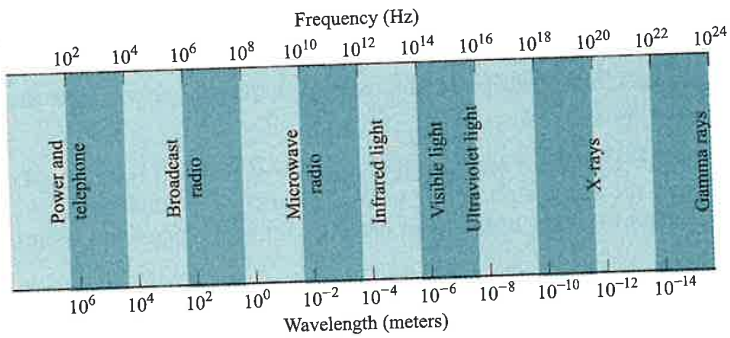


FIGURE 3.35 Electromagnetic spectrum

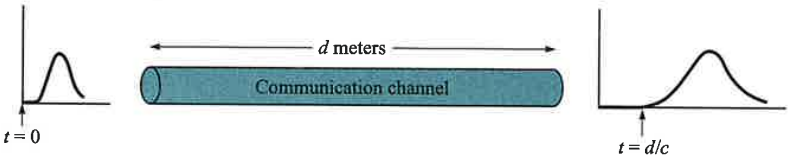


FIGURE 3.36 Propagation delay of a pulse over the communication channel

making the medium broadcast in nature. This condition leads to a network topology that is *continuous* in nature. In addition, all users within receiving range of each other must share the frequency band that is available and can thus interfere with each other. Unlike wired media, the radio spectrum is finite, and it is not possible to procure additional capacity. A given frequency band can be reused only in a sufficiently distant geographical area. To maximize its utility, the radio spectrum is closely regulated by government agencies.

Another difference between wired and wireless media is that wired media require establishing a right-of-way through the land that is traversed by the cable. This process is complicated, costly, and time-consuming. On the other hand, systems that use wireless media do not require the right-of-way and can be deployed by procuring only the sites where the antennas are located. Wireless systems can therefore be deployed more quickly and at lower cost.

Finally, we note that for wired media the attenuation has an exponential dependence on distance; that is, the attenuation at a given frequency is of the form 10^{kd} where the constant k depends on the specific frequency and d is that distance. The attenuation for wired media in dB is then

$$\text{attenuation for wired media} = kd \text{ dB}$$

That is, the attenuation in dB increases linearly with the distance. For wireless media the attenuation is proportional to d^n where n is the **path loss exponent**. For free space $n = 2$, and for environments where obstructions are present $n > 2$. The attenuation for wireless media in dB is then

$$\text{attenuation for wireless media is proportional to } n \log_{10} d \text{ dB}$$

and so the attenuation in dB only increases logarithmically with the distance. Thus in general the signal level in wireless systems can be maintained over much longer distances than in wired systems.

3.7.1 Twisted Pair

The simplest guided transmission medium consists of two parallel insulated conducting (e.g., copper) wires. The signal is transmitted through one wire while a ground reference is transmitted through the other. This two-wire system is susceptible to crosstalk and noise. *Crosstalk* refers to the picking up of electrical signals from other adjacent wires. Because the wires are unshielded, there is

also a tendency to pick up noise, or *interference*, from other electromagnetic sources such as broadcast radio. The receiver detects the information signal by the voltage difference between the ground reference signal and the information signal. If either one is greatly altered by interference or crosstalk, then the chance of error is increased. For this reason parallel two-wire lines are limited to short distances.

A **twisted pair** consists of two wires that are twisted together to reduce the susceptibility to interference. The close proximity of the wires means that any interference will be picked up by both and so the difference between the pair of wires should be largely unaffected by the interference. Twisting also helps to reduce (but not eliminate) the crosstalk interference when multiple pairs are placed within one cable. Much of the wire in the telephone system is twisted-pair wire. For example, it is used between the customer and the central office, also called the *subscriber loop*, and often between central offices, called the *trunk plant*. Because multiple pairs are bundled together within one telephone cable, the amount of crosstalk is still significant, especially at higher frequencies.

A twisted pair can pass a relatively wide range of frequencies. The attenuation for twisted pair, measured in decibels/mile, can range from a 1 to 4 dB/mile at 1 kHz to 10 to 20 dB/mile at 500 kHz, depending on the gauge (diameter) of the wire as shown in Figure 3.37. Thus the bandwidth of twisted pair decreases with distance. Table 3.5 shows practical limits on data rates that can be achieved in a unidirectional link over a 24-gauge (0.016-inch-diameter wire) twisted pair for various distances.

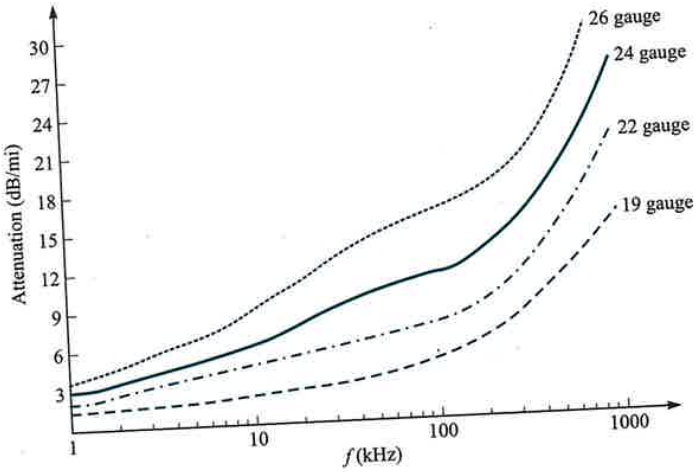


FIGURE 3.37 Attenuation versus frequency for twisted pair [after Smith 1985]

Standard	Data rate	Distance
T-1	1.544 Mbps	18,000 feet, 5.5 km
DS2	6.312 Mbps	12,000 feet, 3.7 km
1/4 STS-1	12.960 Mbps	4500 feet, 1.4 km
1/2 STS-1	25.920 Mbps	3000 feet, 0.9 km
STS-1	51.840 Mbps	1000 feet, 300 m

TABLE 3.5 Data rates of 24-gauge twisted pair

The first digital transmission system used twisted pair and was used in the trunk portion of the telephone network. This T-1 carrier system achieved a transmission rate of 1.544 Mbps and could carry 24 voice channels. The T-1 carrier system used baseband pulse transmission with bipolar encoding. The T-1 carrier system is discussed further in Chapter 4 in the context of the telephone network. Twisted pair in the trunk portion of the telephone network is being replaced by optical fibers. However, twisted pair constitutes the bulk of the access network that connects users to the telephone office, and as such, is crucial to the evolution of future digital networks. In the remainder of this section, we discuss how new systems are being introduced to provide high-speed digital communications in the access network.

Originally, in optimizing for the transmission of speech, the telephone company elected to transmit frequencies within the range of 0 to 4 kHz. Limiting the frequencies at 4 kHz reduced the crosstalk that resulted between different cable pairs at the higher frequencies and provided the desired voice quality. Within the subscriber loop portion, **loading coils** were added to further improve voice transmission within the 3 kHz band by providing a flatter transfer function. This loading occurred in lines longer than 5 kilometers. While improving the quality of the speech signal, the loading coils also increased the attenuation at the higher frequencies and hence reduced the bandwidth of the system. Thus the choice of a 4 kHz bandwidth for the voice channel and the application of loading coils, not the inherent bandwidth of twisted pair, are the factors that limit digital transmission over telephone lines to approximately 40 kbps.

Application—Digital Subscriber Loops

Several digital transmission schemes were developed in the 1970s to provide access *to Integrated Services Digital Network (ISDN)* using the twisted pair (without loading coils) in the subscriber loop network. These schemes provide for two Bearer (B) 64 kbps channels and one data (D) 16 kbps channel from the user to the telephone network. These services were never deployed on a wide scale.

To handle the recent demand from consumers for higher speed data transmission, the telephone companies plan to introduce a new technology called **asymmetric digital subscriber line (ADSL)**. The objective of this technology is to use existing twisted-pair lines to provide the higher bit rates that are possible with unloaded twisted pair. The frequency spectrum is divided into two regions.

Ex. 1015.66

The lower frequencies are used for conventional analog telephone signals. The region above is used for bidirectional digital transmission. The system is asymmetric in that the user can transmit upstream into the network at speeds ranging from 64 kbps to 640 kbps but can receive information from the network at speeds from 1.536 Mbps to 6.144 Mbps, depending on the distance from the telephone central office. This asymmetry in upstream/downstream transmission rates is said to match the needs of current applications such as upstream requests and downstream page transfers in the World Wide Web application.

The ITU-T G.992.1 standard for ADSL uses the Discrete Multitone (DMT) system that divides the available bandwidth into a large number of small subchannels. The binary information is distributed among the subchannels, each of which uses QAM. DMT can adapt to line conditions by avoiding subchannels with poor SNR. ITU-T has also approved standard G.992.2 as a "lite" version that provides access speeds of up to 512 kbps from the user and download speeds of up 1.5 Mbps. The latter is the simpler and less expensive standard because it does not require a "splitter" to separate telephone voice signals from the data signal and can instead be plugged directly into the PC by the user as is customary for most voiceband modems.

Application—Local Area Networks

Twisted pair is installed during the construction of most office buildings. The wires that terminate at the wall plate in each office are connected to wiring closets that are placed at various locations in the building. Consequently, twisted pair is a good candidate for use in local area computer networks where the maximum distance between a computer and a network device is in the order of 100 meters. As a transmission medium, however, high-speed tranmission over twisted pairs poses serious challenges. Several categories of twisted-pair cable have been defined for use in LANs. Category 3 **unshielded twisted pair (UTP)** corresponds to ordinary voice-grade twisted pair and can be used at speeds up to 16 Mbps. Category 5 UTP is intended for use at speeds up to 100 Mbps. Category 5 twisted pairs are twisted more tightly than are those in category 3, resulting in much better crosstalk immunity and signal quality. *Shielded twisted pair* involves providing a metallic braid or sheath to cover each twisted pair. It provides better performance than UTP but is more expensive and more difficult to use.

10BASE-T ETHERNET LAN

The most widely deployed version of Ethernet LAN uses the 10BASE-T physical layer. The designation 10BASE-T denotes *10* Mbps operation using *base*band transmission over *t*wisted-pair wire. The NIC card in each computer is connected to a hub in a star topology as shown in Figure 3.38. Two category 3 UTP cables provide the connection between computer and hub. The transmissions use Manchester line coding, and the cables are limited to a maximum distance of 100 meters.

Ex. 1015.67

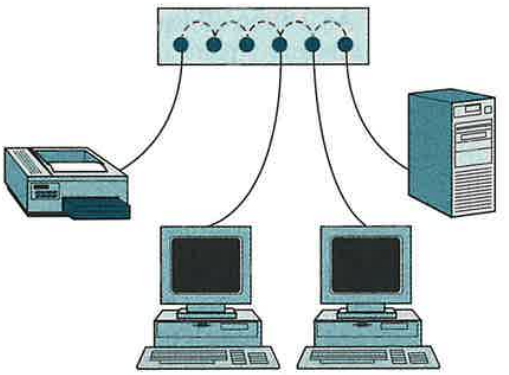


FIGURE 3.38 Ethernet hub

100BASE-T ETHERNET LAN

The 100BASE-T Ethernet LAN is also known as *Fast Ethenet*. As indicated by the designation, 100BASE-T Ethernet operates at a speed of 100 Mbps using twisted-pair wire. The computers are connected to a *hub* or a *switch* in a star topology, and the distance of the twisted pairs is limited to 100 meters. Operating 100 Mbps on UTP is challenging, and so three options for doing so were developed, one for category 3 UTP, one for shielded twisted pair, and one for category 5 UTP. One problem with extending the 10BASE-T transmission format is that Manchester line coding is inefficient in its use of bandwidth. Recall from the section on line coding that Manchester coding pulses vary at twice the information rate, so the use of Manchester coding would have required operation at 200 Mpulses/second. Another problem is that higher pulse rates result in more electromagnetic interference. For this reason, new and more efficient line codes were used in the new standards.

In the 100BASE-T4 format, four category 3 twisted-pair wires are used. At any given time three pairs are used to jointly provide 100 Mbps in a given direction; that is, each pair provides 33 1/3 Mbps. The fourth pair is used for collision detection. The transmission uses ternary signaling in which the transmitted pulses can take on three levels, $+A$, 0, or $-A$. The line code maps a group of eight bits into a corresponding group of six ternary symbols that are transmitted over the three parallel channels over two pulse intervals, or equivalently four bits into three ternary symbols/pulse interval. This mapping is possible because $2^4 = 16 < 3^3 = 27$. The transmitter on each pair sends 25 Mpulses/second, which gives a bit rate of 25 Mp/s $\times$ 4 bits/3 pulses = 33 1/3 Mbps as required. As an option, four category 5 twisted pairs can be used instead category 3 twisted pairs.

In the 100BASE-TX format, two category 5 twisted pairs are used to connect to the hub. Transmission is full duplex with each pair transmitting in one of the directions at a pulse rate of 125 Mpulses/second. The line code used takes a group of four bits and maps it into five binary pulses, giving a bit rate of 125

Mpulses/second × 4 bits/pulse = 100 Mbps. An option allows two pairs of shielded twisted wire to be used instead of the category 5 pairs.

3.7.2 Coaxial Cable

In **coaxial cable** a solid center conductor is located coaxially within a cylindrical outer conductor. The two conductors are separated by a solid dielectric material, and the outer conductor is covered with a plastic sheath as shown in Figure 3.39. The coaxial arrangement of the two conductors provides much better immunity to interference and crosstalk than twisted pair does. By comparing Figure 3.40 with Figure 3.37, we can see that coaxial cable can provide much higher bandwidths (hundreds of MHz) than twisted pair (a few MHz). For example, existing cable television systems use a bandwidth of 500 MHz.

Coaxial cable was initially deployed in the backbone of analog telephone networks where a single cable could be used to carry in excess of 10,000 simultaneous analog voice circuits. Digital transmission systems using coaxial cable were also deployed in the telephone network in the 1970s. These systems operate in the range of 8.448 Mbps to 564.992 Mbps. However, the deployment of coaxial cable transmission systems in the backbone of the telephone network was discontinued because of the much higher bandwidth and lower cost of optical fiber transmission systems.

Application—Cable Television Distribution

The widest use of coaxial cable is for distribution of television signals in cable TV systems. Existing coaxial cable systems use the frequency range from 54 MHz to 500 MHz. A National Television Standards Committee (NTSC) analog television signal occupies a 6 MHz band, and a phase alternation by line (PAL) analog television signal occupies 8 MHz, so 50 to 70 channels can be accommodated.[11] Existing cable television systems are arranged in a tree-and-branch topology as shown in Figure 3.41. The master television signal originates at a head end office,

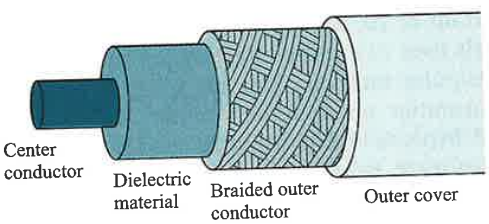


FIGURE 3.39 Coaxial cable

[11]The NTSC and PAL formats are two standards for analog television. The NTSC format is used in North America and Japan, and the PAL format is used in Europe.

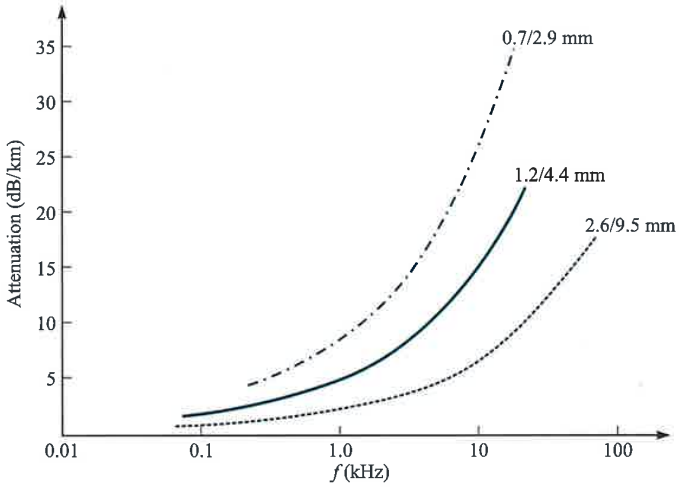


FIGURE 3.40 Attenuation versus frequency for coaxial cable [after Smith 1985]

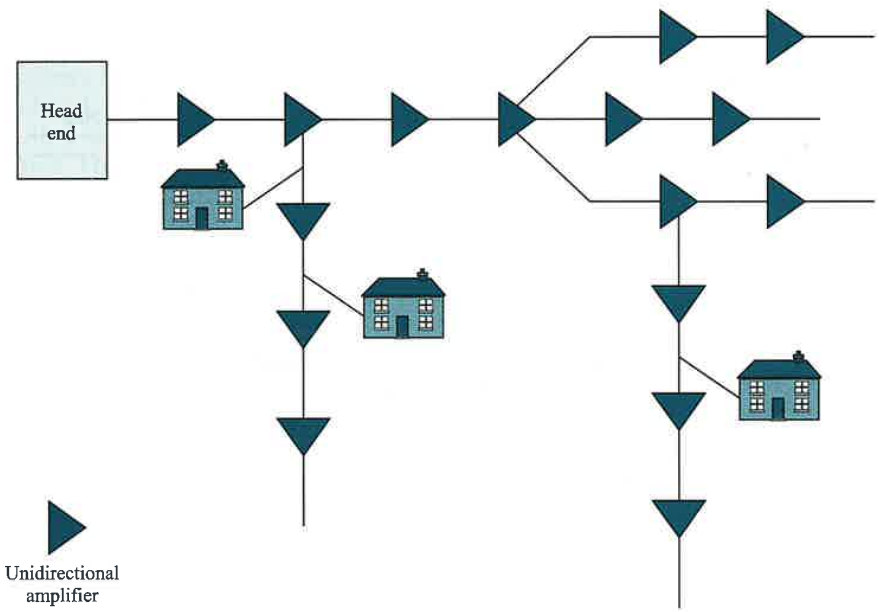


FIGURE 3.41 Tree-and-branch topology of conventional cable TV systems

and unidirectional analog amplifiers maintain the signal level. The signal is split along different branches until all subscribers are reached. Because all the information flows from the head end to the subscribers, cable television systems were designed to be unidirectional.

Application—Cable Modem

The coaxial cable network has a huge bandwidth flowing from the network to the user. For example, a single analog television channel will provide approximately 6 MHz of bandwidth. If QAM modulation is used with a 64-point constellation, then a bit rate of 6 Mpulses/second × 6 bits/pulse = 36 Mbps is possible. However, the coaxial network was not designed to provide communications from the user to the network. Figure 3.42 shows how coaxial cable networks are being modified to provide upstream communications through the introduction of bidirectional split-band amplifiers that allow information to flow in both directions.

Figure 3.43a shows the existing cable spectrum that uses the band from 54 MHz to 500 MHz for the distribution of analog television signals. Figure 3.43b shows the proposed spectrum for hybrid fiber-coaxial systems. The band from 550 MHz to 750 MHz would be used to carry new digital video and data signals as well as downstream telephone signals. In North America channels are 6 MHz wide, so these downstream channels can support bit rates in the range of 36 Mbps. The band from 5 MHz to 42 MHz, which was originally intended for

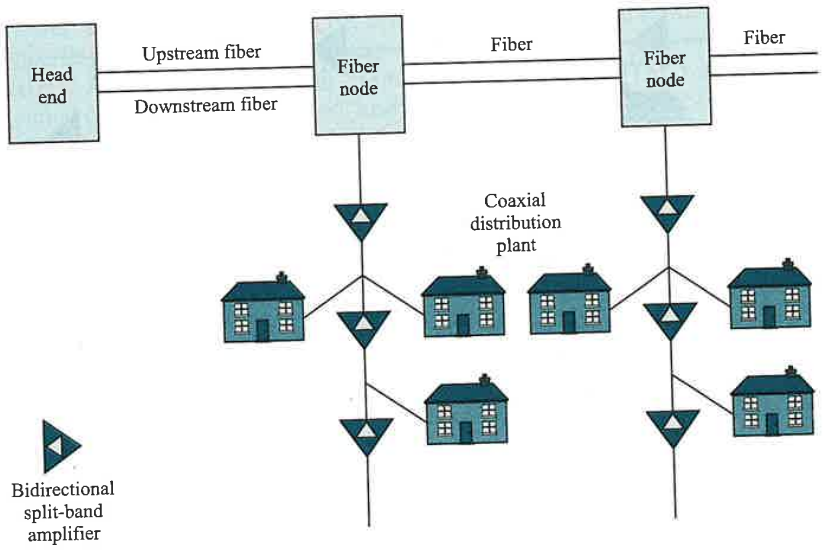


FIGURE 3.42 Topology of hybrid fiber-coaxial systems

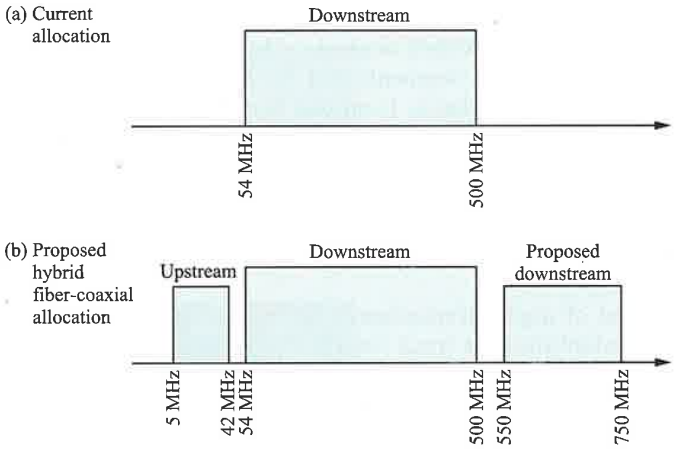


FIGURE 3.43 Frequency allocation in cable TV systems

pay-per-view signaling, would be converted for **cable modem** upstream signals as well as for *cable telephony*. This lower band is subject to much worse interference and noise than the downstream channels. Using channels of approximately 2 MHz, upstream transmission rates from 500 kbps to 4 Mbps could be provided. As in the case of ADSL, we see that the upstream/downstream transmission rates are asymmetric.

Both the upstream and downstream channels need to be *shared* among subscribers in the feeder line. The arrangement is similar to that of a local area network in that the cable modems from the various users must communicate with a *cable modem termination system (CMTS)* at the operator's end of the cable. The cable modems must listen for packets destined to them on an assigned downstream channel. They must also contend to obtain time slots to transmit their information in an assigned channel in the upstream direction.

Application—Ethernet LAN

The original design of the Ethernet LAN used coaxial cable for the shared medium (see Figure 1.17a.) Coaxial cable was selected because it provided high bandwidth, offered good noise immunity, and led to a cost-effective transceiver design. The original standard specified 10Base5, which uses thick (10 mm) coaxial cable operating at a data rate of *10* Mbps, using *base*band transmission and with a maximum segment length of 500 meters. The transmission uses Manchester coding. This cabling system required the use of a *transceiver* to attach the NIC card to the coaxial cable. The thick coaxial cable Ethernet was typically deployed along the ceilings in building hallways, and a connection from a workstation in an office would tap onto the cable. Thick coaxial cable is awkward to handle and install. The 10Base2 standard uses thin (5 mm) coaxial

cable operating 10 Mbps and with a maximum segment of 185 meters. The cheaper and easier to handle thin coaxial cable makes use of T-shaped connectors. 10Base5 and 10Base2 segments can be combined through the use of a *repeater* that forwards the signals from one segment to the other.

3.7.3 Optical Fiber

The deployment of digital transmission systems using twisted pair and coaxial cable systems established the trend toward digitization of the telephone network during the 1960s and 1970s. These new digital systems provided significant economic advantages over previous analog systems. Optical fiber transmission systems, which were introduced in the 1970s, offered even greater advantages over copper-based digital transmission systems and resulted in a dramatic acceleration of the pace toward digitization of the network. Figure 1.9 of Chapter 1 showed that optical fiber systems represented an acceleration in the long-term rate of improvement in transmission capacity.

The typical T-1 or coaxial transmission system requires repeaters about every 2 km. Optical fiber systems, on the other hand, have maximum repeater spacings in the order of tens to hundreds of kilometers. The introduction of optical fiber systems has therefore resulted in great reductions in the cost of digital transmission. Optical fiber systems have also allowed dramatic reductions in the space required to house the cables. A single fiber strand is much thinner than twisted pair or coaxial cable. Because a single optical fiber can carry much higher transmission rates than copper systems, a single cable of optical fibers can replace many cables of copper wires. In addition, optical fibers do not radiate significant energy and do not pick up interference from external sources. Thus compared to electrical transmission, optical fibers are more secure from tapping and are also immune to interference and crosstalk.

Optical fiber consists of a very fine cylinder of glass (core) surrounded by a concentric layer of glass (cladding) as shown in Figure 3.44. The information itself is transmitted through the core in the form a fluctuating beam of light. The core has a slightly higher optical density (index of refraction) than the cladding. The ratio of the indices of refraction of the two glasses defines a critical angle θ_c. When a ray of light from the core approaches the cladding at an angle less than θ_c, the ray is completely reflected back into the core. In this manner the ray of light is guided within the fiber.

The attenuation in the fiber can be kept low by controlling the impurities that are present in the glass. When it was invented in 1970, optical fiber had a loss of 20 dB per kilometer. Within 10 years systems with a loss of 0.2 dB/km had become available. Figure 3.45 shows that minimum attenuation of optical fiber varies with the wavelength of the signal. It can be seen that the systems that operate at wavelengths of 850 nanometer (nm), 1300 nm, and 1550 nm occupy regions of low attenuation. The attenuation peaks are due to residual water

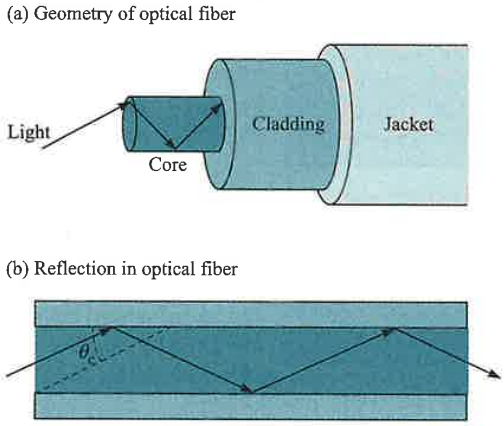


FIGURE 3.44 Transmission of light waves in optical fiber

vapor in the glass fiber. Early optical fiber transmission systems operated in the 850 nm region at bit rates in the tens of megabits per second and used relatively inexpensive light emitting diodes (LEDs) as the light source. Second- and third-generation systems use laser sources and operate in the 1300 nm and 1500 nm region achieving gigabits/second bit rates.

A **multimode fiber** has an input ray of light reach the receiver over multiple paths, as shown in Figure 3.46a. Here the first ray arrives in a direct path, and the second ray arrives through a reflected path. The difference in delay between the two paths causes the rays to interfere with each other. The amount of interference depends on the duration of a pulse relative to the delays of the

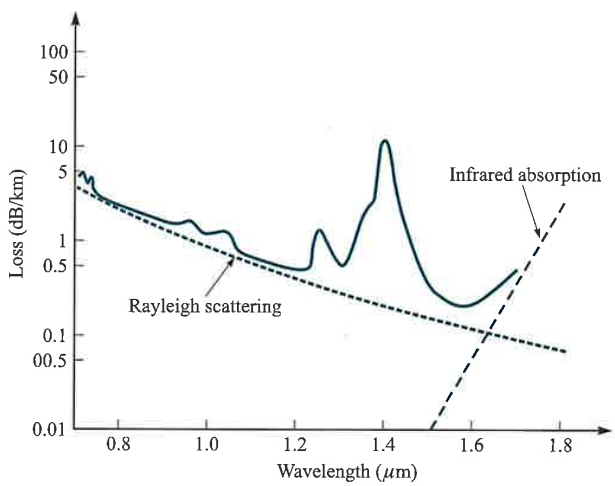


FIGURE 3.45 Attenuation versus wavelength for optical fiber

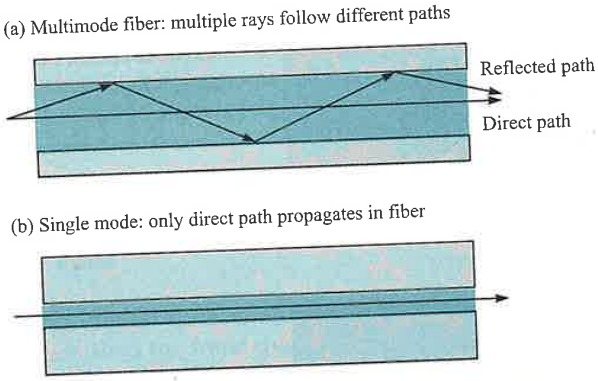


FIGURE 3.46 Single-mode and multimode optical fiber

paths. The presence of multiple paths limits the maximum bit rates that are achievable using multimode fiber. By making the core of the fiber much narrower, it is possible to restrict propagation to the single direct path. These **single-mode fibers** can achieve speeds of many gigabits/second over hundreds of kilometers.

Figure 3.47 shows an optical fiber transmission system. The transmitter consists of a light source that can be modulated according to an electrical input signal to produce a beam of light that is inserted into the fiber. Typically the binary information sequence is mapped into a sequence of on/off light pulses at some particular wavelength. An optical detector at the receiver end of the system converts the received optical signal into an electrical signal from which the original information can be detected.

The region around 1300 nm contains a band with attenuation less than 0.5 dB/km. This region has a bandwidth of 25 terahertz. One terahertz is 10^{12} Hz, that is, 1 million MHz! The region around 1550 nm has another band with attenuation as low as 0.2 dB/km [Mukherjee 1997]. This region has a bandwidth of about 25 THz. Clearly, existing optical transmission systems do not come close to utilizing this bandwidth.

Wavelength-division multiplexing (WDM) is one approach for attempting to exploit more of the available bandwidth. In WDM multiple wavelengths are used

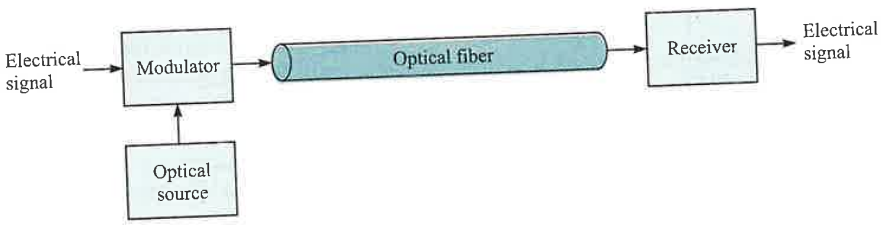


FIGURE 3.47 Optical transmission system

to simultaneously carry several information streams over the same fiber. WDM is a form of multiplexing and is covered in Chapter 4. Early WDM systems handled 16 wavelengths each transmitting 2.5 Gbps for a total of 40 Gbps. These systems could span distances in excess of 300 km. Dense WDM systems can provide 160 wavelengths, each operating at 10 Gbps for a total of 1600 Gbps.

As the light pulses propagate through the fiber, the pulses become spread out. This dispersion limits the minimum time between consecutive pulses and hence the bit rate. A special pulse shape, called a *soliton*, that retains its shape as it propagates through the fiber has been discovered. Experiments have demonstrated that solitons can achieve speeds of 80 Gbps over distances of 10,000 km. Early field trials have demonstrated soliton transmission at 10 Gbps over 200 km. Soliton-based systems promise extremely high-speed repeaterless digital transmission systems.

Application—Network Backbone

Optical fiber transmission systems are widely deployed in the backbone of networks. In Chapter 4 we present the digital multiplexing hierarchy that has been developed for electrical and optical digital transmission systems. Current optical fiber transmissions sytems provide transmission rates from 45 Mbps to 9.6 Gbps using single wavelength transmission and 40 Gbps to 1600 Gbps using WDM. Optical fiber systems are very cost-effective in the backbone of networks because the cost is spread over a large number of users. Optical fiber transmissions systems provide the facilities for long-distance telephone communications. Repeaterless optical fiber transmission systems are also used to interconnect telephone offices in metropolitan areas.

The cost of installing optical fiber in the subscriber portion of the network remains higher than the cost of using the existing installed base of twisted pair and coaxial cable. Fiber-to-the-home proposals that would provide huge bandwidths to the user remain too expensive. Fiber-to-curb proposals attempt to reduce this cost by installing fiber to a point that is sufficiently close to the subscriber that twisted pair or coaxial cable can provide high data rates to the user.

Application—Local Area Networks

Optical fiber is used as the physical layer of several LAN standards. The 10BASE-FP Ethernet physical layer standard uses optical fiber operating with an 850 nm source. The transmission system uses Manchester coding and intensity-light modulation and allows distances up to 2 km. The Fiber Distributed Data Interface (FDDI) ring-topology LAN uses optical fiber transmission at a speed of 100 Mbps, using LED light sources at 1300 nm with repeater spacings of up to 2 km. The binary information is encoded using a 4B5B code followed by NRZ inverted line coding. The 100BASE-FX Fast Ethernet physical layer standard uses two fibers, one for send and one for receive. The maximum distance is

Ex. 1015.76

limited to 100 meters. The transmission format is the same as that of FDDI with slight modifications.

Optical fiber is the preferred medium for Gigabit Ethernet. As has become the practice in the development of physical layer standards, the 1000BASE-X standards are based on the preexisting fiber channel standard. The pulse transmission rate is 1.25 gigapulses/second, and an 8B10B code is used to provide the 1 Gbps transmission rate. There are two variations of the 1000BASE-X standard. The 1000BASE-SX uses a "shortwave" light source, nominally 850 nm, and multimode fiber. The distance limit is 550 meters. The 1000BASE-LX uses a "longwave" light source, nominally at 1300 nm, single mode or multimode fiber. For multimode fiber the distance limit is 550 meters, and for single mode fiber the distance is 5 km.

3.7.4 Radio Transmission

Radio encompasses the electromagnetic spectrum in the range of 3 kHz to 300 GHz. In radio communications the signal is transmitted into the air or space, using an antenna that radiates energy at some carrier frequency. For example, in QAM modulation the information sequence determines a point in the signal constellation that specifies the amplitude and phase of the cosine wave that is transmitted. Depending on the frequency and the antenna, this energy can propagate in either a unidirectional or omnidirectional fashion. In the unidirectional case a properly aligned antenna receives the modulated signal, and an associated receiver in the direction of the transmission recovers the original information. In the omnidirectional case any receiver with an antenna in the area of coverage can pick up the signal.

Radio communication systems are subject to a variety of transmission impairments. We indicated earlier that the attenuation in radio links varies logarithmically with the distance. Attenuation for radio systems also increases with rainfall. Radio systems are subject to multipath fading and interference. Multipath fading refers to the interference that results at a receiver when two or more versions of the same signal arrive at slightly different times. If the arriving signals differ in polarity, then they will cancel each other. Multipath fading can result in wide fluctuations in the amplitude and phase of the received signal. Interference refers to energy that appears at the receiver from sources other than the transmitter. Interference can be generated by other users of the same frequency band or by equipment that inadvertently transmits energy outside its band and into the bands of adjacent channels. Interference can seriously affect the performance of radio systems, and for this reason regulatory bodies apply strict requirements on the emission properties of electronic equipment.

Figure 3.48 gives the range of various frequency bands and their applications. The frequency bands are classified according to wavelengths. Thus the low frequency (LF) band spans the range 30 kHz to 300 kHz, which corresponds to a

Ex. 1015.77

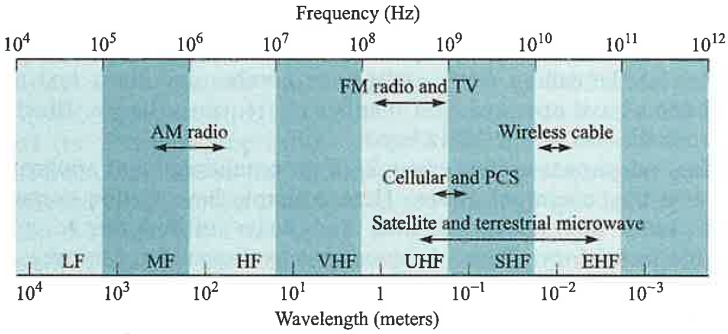


FIGURE 3.48 Radio spectra

wavelength of 1 km to 10 km, whereas the extremely high frequency (EHF) band occupies the range from 30 to 300 GHz corresponding to wavelengths of 1 millimeter to 1 centimeter. Note that the progression of frequency bands in the logarithmic frequency scale have increasingly larger bandwidths, for example, the "band" from 10^{11} to 10^{12} Hz has a bandwidth of 0.9×10^{12} Hz, whereas the band from 10^5 to 10^6 Hz has a bandwidth of 0.9×10^6 Hz.

The propagation properties of radio waves vary with the frequency. Radio waves at the VLF, LF, and MF bands follow the surface of the earth in the form of ground waves. VLF waves can be detected at distances up to about 1000 km, and MF waves, for example, AM radio, at much shorter distances. Radio waves in the HF band are reflected by the ionosphere and can be used for long-distance communications. These waves are detectable only within certain specific distances from the transmitter. Finally, radio waves in the VHF band and higher are not reflected back by the ionosphere and are detectable only within line-of-sight.

In general, radio frequencies below 1 GHz are more suitable for omnidirectional applications, such as those shown in Table 3.6. For example, paging systems ("beepers") are an omnidirectional application that provides one-way communications. A high-power transmission system is used to reach simple, low-power pocket-size receivers in some geographic area. The purpose of the system is to alert the owner that someone wishes to communicate with him or her. The

System	Description	Distance
Paging	Short message	10s of kilometers
Cordless telephone	Analog/digital voice	10s of meters
Cellular telephone	Analog/digital voice and data	kilometers
Personal Communication Services	Digital voice and data	100s of meters
Wireless LAN	High-speed data	100 meters

TABLE 3.6 Examples of omnidirectional systems

system may consist of a single high-powered antenna, or a network of interconnected antennas, or be a nationwide satellite-based transmission system. These systems deliver the calling party's telephone number and short text messages. Paging systems have operated in a number of frequency bands. Most systems currently use the 930 to 932 MHz band.

Cordless telephones are an example of an omnidirectional application that provides two-way communications. Here a simple base station connects to a telephone outlet and relays signaling and voice information to a cordless phone. This technology allows the user to move around in an area of a few tens of meters while talking on the phone. The first generation of cordless phones used analog radio technology and subsequent generations have used digital technology.

Application—Cellular Communications

Analog cellular telephone systems were introduced in 1979 in Japan. This system provided for 600 two-way channels in the 800 MHz band. In Europe the Nordic Mobile Telephone system was developed in 1981 in the 450 MHz band. The U.S. Advanced Mobile Phone System (AMPS) was deployed in 1983 in a frequency band of 50 MHz in the 800 MHz region. This band is divided into 30 kHz channels that can each carry a single FM-modulated analog voice signal.

Analog cellular phones quickly reached their capacity in large metropolitan areas because of the popularity of cellular phone service. Several digital cellular telephone systems based on digital transmission have been introduced. In 1991 Interim Standard IS-54 in the United States allowed for the replacement of a 30 kHz channel with a digital channel that can support three users. This digital channel uses differential QAM modulation in place of the analog GM modulation. A cellular standard based on code division multiple access (CDMA) was also standardized as IS-95. This system, based on direct sequence spread spectrum transmission, can handle more users than earlier systems could. In Europe the Global System for Mobile (GSM) standard was developed to provide for a pan-European digital cellular system in the 900 MHz band. These cellular systems are discussed further in Chapter 6.

In 1995 personal communication services (PCS) licenses were auctioned in the U.S. for spectrum in the 1800/1900 MHz region. PCS is intended to extend digital cellular technology to a broader community of users by using low-power transmitters that cover small areas, "microcells." PCS thus combines aspects of conventional cellular telephone service with aspects of cordless telephones. The first large deployment of PCS is in the Japanese Personal Handiphone system that operates in the 1800/1900 band. This system is now very popular. In Europe the GSM standard has been adapted to the 1800/1900 band.

Application—Wireless LANs

Wireless LANs are another application of omnidirectional wireless communications. The objective here is to provide high-speed communications among a

Ex. 1015.79

number of computers located in relatively close proximity. Most standardization efforts in the United States have focused in the Industrial/Scientific/Medical (ISM) bands, which span 902 to 928 MHz, 2400 to 2483.5 MHz, and 5725 to 5850 MHz, respectively. Unlike other frequency bands, the ISM band is designated for unlicensed operation so each user must cope with the interference from other users. In Europe, the high-performance radio LAN (HIPERPLAN) standard was developed to provide high-speed (20 Mbps) operation in the 5.15 to 5.30 GHz band. In 1996 the Federal Communications Commission (FCC) in the United States announced its intention to make 350 MHz of spectrum in the 5.15 to 5.35 GHz and 5.725 to 5.825 GHz bands available for unlicensed use in LAN applications. These developments are significant because these systems will provide high-speed communications to the increasing base of portable computers. This new spectrum allocation will also enable the development of ad hoc digital radio networks in residential and other environments.

Application—Point-to-Point and Point-to-Multipoint Radio Systems

Highly directional antennas can be built for microwave frequencies that cover the range from 2 to 40 GHz. For this reason point-to-point wireless systems use microwave frequencies and were a major component of the telecommunication infrastructure introduced several years ago. Digital microwave transmission systems have been deployed to provide long-distance communications. These systems typically use QAM modulation with fairly large signal constellations and can provide transmission rates in excess of 100 Mbps. The logarithmic, rather than linear, attenuation gave microwave radio systems an advantage over coaxial cable systems by requiring repeater spacings in the tens of kilometers. In addition, microwave systems did not have to deal with right-of-way issues. Microwave transmission systems can also be used to provide inexpensive digital links between buildings.

Microwave frequencies in the 28 GHz band have also been licensed for point-to-multipoint "wireless cable" systems. In these systems microwave radio beams from a telephone central office would send 50 Mbps directional signals to subscribers within a 5 km range. Reflectors would be used to direct these beams so that all subscribers can be reached. These signals could contain digital video and telephone as well as high-speed data. Subscribers would also be provided with transmitters that would allow them to send information upstream into the network. The providers of this service have about 1 GHz in total bandwidth available.

Application—Satellite Communications

Early satellite communications systems can be viewed as microwave systems with a single repeater in the sky. A (geostationary) satellite is placed at an altitude of about 36,000 km above the equator where its orbit is stationary relative to the rotation of the earth. A modulated microwave radio signal is beamed to the

satellite on an uplink carrier frequency. A transponder in the satellite receives the uplink signal, regenerates it, and beams it down back to earth on a downlink carrier frequency. A satellite typically contains 12 to 20 transponders so it can handle a number of simultaneous transmissions. Each transponder typically handles about 50 Mbps. Satellites operate in the 4/6, 11/14, and 20/30 GHz bands, where the first number indicates the downlink frequency and the second number the uplink frequency.

Geostationary satellite systems have been used to provide point-to-point digital communications to carry telephone traffic between two points. Satellite systems have an advantage over fiber systems in situations where communications needs to be established quickly or where deploying the infrastructure is too costly. Satellite systems are inherently broadcast in nature, so they are also used to simultaneously beam television, and other signals, to a large number of users. Satellite systems are also used to reach mobile users who roam wide geographical areas.

Constellations of low-earth orbit satellites (LEOS) are being planned for deployment. These include the Iridium and Teledesic systems. The satellites are not stationary with respect to the earth, but they rotate in such a way that there is continuous coverage of the earth. The component satellites are interconnected by high-speed links forming a network in the sky.

3.7.5 Infrared Light

Infrared light is a communication medium whose properties differ significantly from radio frequencies. Infrared light does not penetrate walls, so an inherent property is that it is easily contained within a room. This factor can be desirable from the point of view of reducing interference and enabling reuse of the frequency band in different rooms. Infrared communications systems operate in the region from 850 nm to 900 nm where receivers with good sensitivity are available. Infrared light systems have a very large potential bandwidth that is not yet exploited by existing systems. A serious problem is that the sun generates radiation in the infrared band, which can be a cause of severe interference. The infrared band is being investigated for use in the development of very high speed wireless LANs.

Application—IrDA Links

The Infrared Data Association (IrDA) was formed to promote the development of infrared light communication systems. A number of standards have been developed under its auspices. The IrDA-C standard provides bidirectional communications for cordless devices such as keyboards, mice, joysticks, and handheld computers. This standard operates at a bit rate of 75 kbps at distances of up to 8 meters. The IrDA-D standard provides for data rates from 115 kb/s to

Ex. 1015.81

4 Mb/s over a distance of 1 meter. It was designed as a wireless alternative to connecting devices, such as a laptop to a printer.

3.8 ERROR DETECTION AND CORRECTION

In most communication channels a certain level of noise and interference is unavoidable. Even after the design of the digital transmission system has been optimized, bit errors in transmission will occur with some small but nonzero probability. For example, typical bit error rates for systems that use copper wires are in the order of 10^{-6}, that is, one in a million. Modern optical fiber systems have bit error rates of 10^{-9} or less. In contrast, wireless transmission systems can experience error rates as high as 10^{-3} or worse. The acceptability of a given level of bit error rate depends on the particular application. For example, certain types of digital speech transmission are tolerant to fairly high bit error rates. Other types of applications such as electronic funds transfer require essentially error-free transmission. In this section we introduce **error-control** techniques for improving the error-rate performance that is delivered to an application in situations where the inherent error rate of a digital transmission system is unacceptable.

There are two basic approaches to error control. The first approach involves the detection of errors and an **automatic retransmission request (ARQ)** when errors are detected. This approach presupposes the availability of a return channel over which the retransmission request can be made. For example, ARQ is widely used in computer communication systems that use telephone lines. The second approach, **forward error correction (FEC)**, involves the detection of errors followed by processing that attempts to correct the errors. FEC is appropriate when a return channel is not available, retransmission requests are not easily accommodated, or a large amount of data is sent and retransmission to correct a few errors is very inefficient. For example, FEC is used in satellite and deep-space communications. A recent application is in audio CD recordings where FEC is used to provide tremendous robustness to errors so that clear sound reproduction is possible even in the presence of smudges and scratches on the disk surface. Error detection is the first step in both ARQ and FEC. The difference between ARQ and FEC is that ARQ "wastes" bandwidth by using retransmissions, whereas FEC requires additional redundancy in the transmitted information and incurs significant processing complexity in performing the error correction.

In this section we discuss parity check codes, the Internet checksum, and polynomial codes that are used in error detection. We also present methods for assessing the effectiveness of these codes in several error environments. These results are used in Chapter 5, in the discussion of ARQ protocols. An optional section on linear codes gives a more complete introduction to error detection and correction.

Ex. 1015.82

3.8.1 Error Detection

In this section we discuss the idea of error detection in general terms, using the single parity check code as an example throughout the discussion. The basic idea in performing error detection is very simple. As illustrated in Figure 3.49, the information produced by an application is encoded so that the stream that is input into the communication channel satisfies a specific *pattern* or condition. The receiver checks the stream coming out of the communication channel to see whether the pattern is satisfied. If it is not, the receiver can be certain that an error has occurred and therefore sets an alarm to alert the user. This certainty stems from the fact that no such pattern would have been transmitted by the encoder.

The simplest code is the **single parity check code** that takes k information bits and appends a single **check bit** to form a **codeword**. The parity check ensures that the total number of 1s in the codeword is even; that is, the codeword has even parity.[12] The check bit in this case is called a *parity bit*. This error-detection code is used in ASCII where characters are represented by seven bits and the eighth bit consists of a parity bit. This code is an example of the so-called linear codes because the parity bit is calculated as the modulo 2 sum of the information bits:

$$b_{k+1} = b_1 + b_2 + \ldots + b_k \quad \text{modulo } 2 \tag{2}$$

where $b_1, b_2, \ldots, b_k$ are the information bits.

Recall that in modulo 2 arithmetic $0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1$, and $1 + 1 = 0$. Thus, if the information bits contain an even number of 1s, then the parity bit will be 0; and if they contain an odd number, then the parity bit will be 1. Consequently, the above rule will assign the parity bit a value that will produce a *codeword* that *always contains an even number of 1s*. This pattern defines the single parity check code.

If a codeword undergoes a single error during transmission, then the corresponding binary block at the output of the channel will contain an odd number of 1s and the error will be detected. More generally, if the codeword undergoes an odd number of errors, the corresponding output block will also contain an odd number of 1s. Therefore, the single parity bit allows us to detect all error patterns that introduce an odd number of errors. On the other hand, the single parity bit will fail to detect any error patterns that introduce an even number of errors, since the resulting binary vector will have even parity. Nonetheless, the

FIGURE 3.49 General error-detection system

[12]Some systems use odd parity by defining the check bit to be the binary complement of equation 2.

single parity bit provides a remarkable amount of error-detection capability, since the addition of a single check bit results in making half of all possible error patterns detectable, regardless of the value of k.

Figure 3.50 shows an alternative way of looking at the operation of this example. At the transmitter a checksum is calculated from the information bits and transmitted along with the information. At the receiver, the checksum is recalculated, based on the received information. The received and recalculated checksums are compared, and the error alarm is set if they disagree.

This simple example can be used to present two fundamental observations about error detection. The first observation is that error detection requires **redundancy** in that the amount of information that is transmitted is over and above the required minimum. For a single parity check code of length $k+1$, k bits are information bits, and one bit is the parity bit. Therefore, the fraction $1/(k+1)$ of the transmitted bits is redundant.

The second fundamental observation is that *every error-detection technique will fail to detect some errors.* In particular, an error-detection technique will always fail to detect transmission errors that convert a valid codeword into another valid codeword. For the single parity check code, an even number of transmission errors will always convert a valid codeword to another valid codeword.

The objective in selecting an error-detection code is to select the codewords that reduce the likelihood of the transmission channel converting one valid codeword into another. To visualize how this is done, suppose we depict the set of all possible binary blocks as the space shown in Figure 3.51, with codewords shown by xs in the space and noncodewords by os. To minimize the probability of error-detection failure, we want the codewords to be selected so that they are spaced as far away from each other as possible. Thus the code in Figure 3.51a is a poor code because the codewords are close to each other. On the other hand, the code in Figure 3.51b is good because the distance between codewords is maximized. The effectiveness of a code clearly depends on the types of errors that are introduced by the channel. We next consider how the effectiveness is evaluated for the example of the single parity check code.

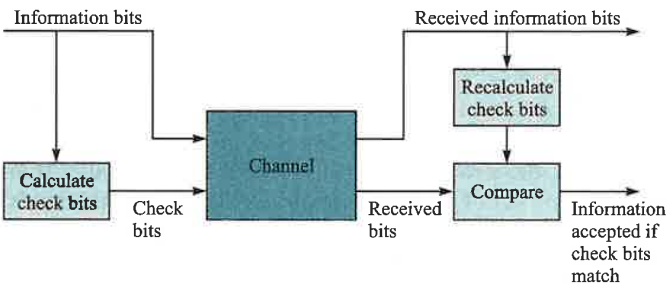


FIGURE 3.50 Error-detection system using check bits

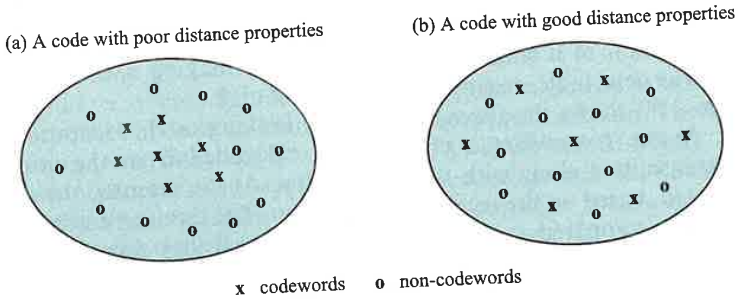


FIGURE 3.51 Distance properties of codes

EFFECTIVENESS OF ERROR-DETECTION CODES

The effectiveness of an error-detection code is measured by the probability that the system fails to detect an error. To calculate this probability of error-detection failure, we need to know the probabilities with which various errors occur. These probabilities depend on the particular properties of the given communication channel. We will consider three models of error channels: the random error vector model, the random bit error model, and burst errors.

Suppose we transmit a codeword that has n bits. Define the error vector $\underline{e} = (e_1, e_2, \ldots, e_n)$ where $e_i = 1$ if an error occurs in the ith transmitted bit and $e_i = 0$ otherwise. In one extreme case, the **random error vector model**, all 2^n possible error vectors are equally likely to occur. In this channel model the probability of $\underline{e}$ does not depend on the number of errors it contains. Thus the error vector $(1, 0, \ldots, 0)$ has the same probability of occurrence as the error vector $(1, 1, \ldots, 1)$. The single parity check code will fail when the error vector has an even number of 1s. Thus for the random error vector channel model, the probability of error detection failure is 1/2.

Now consider the **random bit error model** where the bit errors occur independently of each other. Satellite communications provide an example of this type of channel. Let p be the probability of an error in a single-bit transmission. The probability of an error vector that has j errors is $p^j(1-p)^{n-j}$, since each of the j errors occurs with probability p and each of the $n-j$ correct transmissions occurs with probability $1-p$. By rewriting this probability we obtain:

$$p[\underline{e}] = (1-p)^{n-w(\underline{e})}p^{w(\underline{e})} = (1-p)^n\left(\frac{p}{1-p}\right)^{w(\underline{e})} \tag{3}$$

where the **weight** $w(\underline{e})$ is defined as the number of 1s in $\underline{e}$. For any useful communication channel, the probability of bit error is much smaller than 1, and so $p < 1/2$ and $p/(1-p) < 1$. This implies that for the random bit error channel the probability of $\underline{e}$ decreases as the number or errors (1s) increases; that is, an error pattern with a given number of bit errors is more likely than an error pattern with a larger number of bit errors. Therefore this channel tends to map a transmitted codeword into binary blocks that are clustered around the codeword.

Ex. 1015.85

The single parity check code will fail if the error pattern has an even number of 1s. Therefore, in the random bit error model:

$$\begin{aligned} P[\text{error detection failure}] &= P[\text{undetectable error pattern}] \\ &= P[\text{error patterns with even number of 1s}] \\ &= \binom{n}{2}p^2(1-p)^{n-2} + \binom{n}{4}p^4(1-p)^{n-4} + \ldots \end{aligned} \tag{4}$$

where the number of terms in the sum extends up to the maximum possible even number of errors. In the preceding equation we have used the fact that the number of distinct binary n-tuples with j ones and $n-j$ zeros is given by

$$\binom{n}{j} = \frac{n!}{j!(n-j!)}$$

In any useful communication system, the probability of a single-bit error p is much smaller than 1. We can then use the following approximation: $p^i(1-p)^j \approx p^i(1-pj) \approx p^i$. For example, if $p = 10^{-3}$ then $p^2(1-p)^{n-2} \approx 10^{-6}$ and $p^4(1-p)^{n-4} \approx 10^{-12}$. Thus the probability of detection failure is determined by the first term in equation 4. For example, suppose $n = 32$ and $p = 10^{-4}$. Then the probability of error-detection failure is 5×10^{-6}, a reduction of nearly two orders of magnitude.

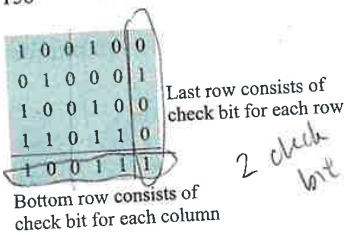
We see then that a wide gap exists in the performance achieved by the two preceding channel models. Many communication channels combine aspects of these two channels in that errors occur in **bursts**. Periods of low error-rate transmission are interspersed with periods in which clusters of errors occur. The periods of low error rate are similar to the random bit error model, and the periods of error bursts are similar to the random error vector model. The probability of error-detection failure for the single parity check code will be between those of the two channel models. In general, measurement studies are required to characterize the statistics of burst occurrence in specific channels.

3.8.2 Two-Dimensional Parity Checks

A simple method to improve the error-detection capability of a single parity check code is to arrange the information bits in columns of k bits, as shown in Figure 3.52. The last bit in each column is the check bit for the information bits in the column. Note that in effect the last column is a "check codeword" over the previous m columns. The right-most bit in each row is the check bit of the other bits in the row. The resulting encoded matrix of bits satisfies the pattern that all rows have even parity and all columns have even parity.

If one, two, or three errors occur anywhere in the matrix of bits during transmission, then at least one row or parity check will fail, as shown in Figure 3.53. However, some patterns with four errors are not detectable, as shown in the figure.

Ex. 1015.86



2 check bit rows/cols

FIGURE 3.52 Two-dimensional parity check code

The two-dimensional parity check code is another example of a linear code. It has the property that error-detecting capabilities can be identified visually, but it does not have particularly good performance. Better codes are discussed in a later (optional) section on linear codes.

3.8.3 Internet Checksum

Several Internet protocols (e.g., IP, TCP, UDP) use check bits to detect errors. With IP a **checksum** is calculated for the contents of the header and included in a special field. Because the checksum must be recalculated at every router, the algorithm for the checksum was selected for its ease of implementation in software rather than for the strength of its error-detecting capabilities.

The algorithm assumes that the header consists of a certain number, say, L, of 16-bit words, $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{L-1}$ plus a checksum $\mathbf{b}_L$. These L words correspond to the "information" in the terminology introduced in the previous sections. The 16-bit checksum $\mathbf{b}_L$ corresponds to the parity bits and is calculated as follows:

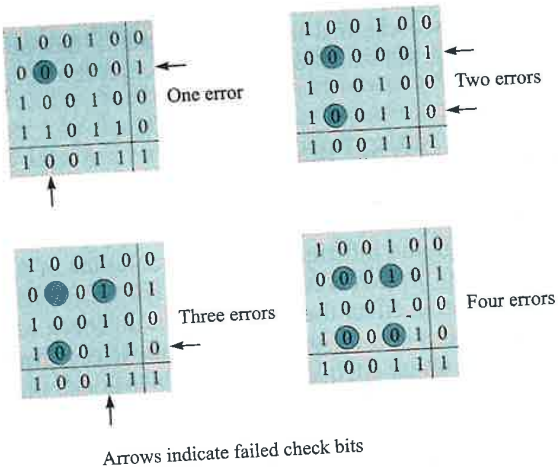


FIGURE 3.53 Detectable and undetectable error patterns for two-dimensional code

1. Each 16-bit word is treated as an integer, and the L words are added modulo $2^{16} - 1$:

$$\mathbf{x} = \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \ldots + \mathbf{b}_{L-1} \text{ modulo } 2^{16} - 1$$

2. The checksum then consists of the negative value of $\mathbf{x}$:

$$\mathbf{b}_L = -\mathbf{x}$$

3. The checksum $\mathbf{b}_L$ is then inserted in a dedicated field in the header.

The contents of all headers, including the checksum field, must then satisfy the following pattern:

$$\mathbf{0} = \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \ldots + \mathbf{b}_{L-1} + \mathbf{b}_L \text{ modulo } 2^{16} - 1$$

Each router can then check for errors in the header by calculating the preceding equation for each received header.

As an example, consider two-bit words with $L = 2$. There are 16 possible combinations for $\mathbf{b}_0$, $\mathbf{b}_1$ and the corresponding checksum $\mathbf{b}_2$. For example if $\mathbf{b}_0 =$ 00 and $\mathbf{b}_1 = 01$, then in modulo 3 arithmetic we have $\mathbf{b}_0 + \mathbf{b}_1 = 0 + 1 = 1$. In modulo 3 arithmetic, $-1 = 2$, so $\mathbf{b}_2 = 2$. The header then satisfies $\mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 = 0 + 1 + 2 = 0$ modulo 3, as required. Note that a sequence of errors will not be detectable if it causes the sum of integers corresponding to the transmitted $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$ to change by a multiple of 3. For example, if we transmit (0, 1, 2) as above but receive (1, 3, 2), then the receiver will calculate $1 + 3 + 2 = 0$ modulo 3, so the errors will not be detected.

The actual Internet algorithm for calculating the checksum is described in terms of 1s complement arithmetic. In this arithmetic, addition of integers corresponds to modulo $2^{16} - 1$ addition, and the negative of the integer corresponding to the 16-bit word $\mathbf{b}$ is found by taking its 1s complement; that is, every 0 is converted to a 1 and vice versa. This process leads to the peculiar situation where there are two representations for $\mathbf{b}$, $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$, which in turn results in additional redundancy in the context of error detection. Given these properties of 1s complement arithmetic, step 1 above then corresponds to simply adding the 16-bit integers $\mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \ldots + \mathbf{b}_{L-1}$ using regular 32-bit addition. The modulo $2^{16} - 1$ reduction is done by taking the 16 higher-order bits in the sum, shifting them down by 16 positions, and adding them back to the sum. Step 2 produces the negative of the resulting sum by taking the 1s complement. Figure 3.54 shows a C function for calculating the Internet checksum adapted from [RFC 1071].

Table 3.7 shows the 16 possible codewords that result when the algorithm is applied to two-bit words with $L = 2$. From the table we can determine the error-detecting capability of this code. The minimum distance of the code is 2; that is, at least two bits are required to change one codeword into another. For example the first two codewords are 000011 (003) and 000110 (012), which differ in the fourth and sixth bit. In general, error patterns that cause the words $\mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \ldots + \mathbf{b}_{L-1} + \mathbf{b}_L$ to change by a total of a multiple of $2^{16} - 1$ will not be detected.

Ex. 1015.88

```
unsigned short cksum(unsigned short *addr, int count)
    {
        /* Compute Internet checksum for "count" bytes
         * beginning at location "addr".
         */
    register long sum = 0;
    while ( count > 1 ) {
        /* This is the inner loop*/
            sum += *addr++;
            count -=2;
        }

        /* Add left-over byte, if any */
    if ( count > 0 )
        sum += *addr;

        /* Fold 32-bit sum to 16 bits */
    while (sum >>16)
        sum = (sum & 0xffff) + (sum >> 16);

    return ~sum;
}
```

FIGURE 3.54 C language program for computing Internet checksum

b_0	b_1	b_2
0	0	3
0	1	2
0	2	1
0	3	0
1	0	2
1	1	1
1	2	0
1	3	2
2	0	1
2	1	0
2	2	2
2	3	1
3	0	0
3	1	2
3	2	1
3	3	0

TABLE 3.7 Resulting codewords when $L = 2$

Ex. 1015.89

3.8.4 Polynomial Codes

We now introduce the class of **polynomial codes** that are used extensively in error detection and correction. Polynomial codes are readily implemented using shift-register circuits and therefore are the most widely implemented error-control codes. Polynomial codes involve generating check bits in the form of a **cyclic redundancy check (CRC)**. For this reason they are also known as CRC codes.

In polynomial codes the information symbols, the codewords, and the error vectors are represented by polynomials with binary coefficients. The k information bits $(i_{k-1}, k_{k-2}, \ldots, i_1, i_0)$ are used to form the **information polynomial** of degree $k-1$:

$$i(x) = i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \ldots + i_1 x + i_0$$

The encoding process takes $i(x)$ and produces a codeword polynomial $b(x)$ that contains the information bits and additional check bits and that satisfies a certain *pattern*. To detect errors, the receiver checks to see whether the pattern is satisfied. Before we explain this process, we need to review polynomial arithmetic.

The polynomial code uses polynomial arithmetic to calculate the codeword corresponding to the information polynomial. Figure 3.55 gives examples of polynomial addition, multiplication, and division using binary coefficients. Note that with binary arithmetic, we have $x^j + x^j = (1+1)x^j = 0$. Note in particular that when the division is completed the remainder $r(x)$ will have a degree smaller than the degree of the divisor polynomial. In the example the divisor polynomial has degree 3, so the division process continues until the remainder term has degree 2 or less.

A polynomial code is specified by its **generator polynomial $g(x)$**. Here we assume that we are dealing with a code in which codewords have n bits, of which k are information bits and $n-k$ are check bits. We refer to this type of

Addition: $(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + (1+1)x^6 + x^5 + 1 = x^7 + x^5 + 1$

Multiplication: $(x+1)(x^2+x+1) = x^3 + x^2 + x + x^2 + x + 1 = x^3 + 1$

Division:

$$\begin{array}{r|l}
 & x^3 + x^2 + x = q(x) \text{ Quotient} \\
x^3 + x + 1 & x^6 + x^5 \quad \leftarrow \text{Dividend} \\
\uparrow & x^6 + \quad x^4 + x^3 \\
\text{Divisor} & \hline
 & x^5 + x^4 + x^3 \\
 & x^5 + \quad x^3 + x^2 \\
 & \hline
 & x^4 + \quad x^2 \\
 & x^4 + \quad x^2 + x \\
 & \hline
 & x = r(x) \text{ Remainder}
\end{array}$$

$$\begin{array}{r}
3 \\
35\overline{)122} \\
105 \\
\hline
17
\end{array}$$

FIGURE 3.55 Polynomial arithmetic

code as an (n, k) code. The generator polynomial for such a code has degree $n-k$ and has the form

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \ldots + g_1 x + 1$$

where $g_{n-k-1}, g_{n-k-2}, \ldots, g_1$ are binary numbers. In the following we will introduce an example that corresponds to a (7,4) code with generator polynomial $g(x) = x^3 + x + 1$.

The calculation of the cyclic redundancy check bits is described in Figure 3.56. First the information polynomial is multiplied by x^{n-k}.

$$x^{n-k}i(x) = i_{k-1}x^{n-1} + i_{k-2}x^{n-2} + \ldots + i_1 x^{n-k+1} + i_0 x^{n-k}$$

If you imagine that the k information bits are in the lower k positions in a register of length n, the multiplication moves the information bits to the k highest-order positions, since the highest term of $i(x)$ can have degree $n-1$. This situation is shown in the example in Figure 3.57. The information polynomial is $i(x) = x^3 + x^2$, so the first step yields $x^3 i(x) = x^6 + x^5$. After three shifts to the left, the contents of the shift register are (1,1,0,0,0,0,0).

Step 2 involves dividing $x^{n-k}i(x)$ by $g(x)$ to obtain the remainder $r(x)$. The terms involved in division are related by the following expression:

$$x^{n-k}i(x) = g(x)q(x) + r(x)$$

The remainder polynomial $r(x)$ provides the CRCs. In the example in Figure 3.57, we have $x^6 + x^5 = g(x)(x^3 + x^2 + x) + x$; that is, $r(x) = x$. In the figure we also show a more compact way of doing the division without explicitly writing the powers of x.

The final step in the encoding procedure obtains the binary codeword $b(x)$ by adding the remainder $r(x)$ from $x^{n-k}i(x)$:

$$b(x) = x^{n-k}i(x) + r(x).$$

Because the divisor $g(x)$ had degree $n-k$, the remainder $r(x)$ can have maximum degree $n-k-1$ or lower. Therefore $r(x)$ has at most $n-k$ terms. In terms of the previously introduced register of length n, $r(x)$ can occupy the lower $n-k$ positions. Recall that the upper k positions were occupied by the

Steps:

1. Multiply $i(x)$ by x^{n-k} (puts zeros in $(n-k)$ low-order positions).

2. Divide $x^{n-k}i(x)$ by $g(x)$.

$$x^{n-k}i(x) = g(x)\,\underbrace{q(x)}_{\text{Quotient}} = \underbrace{r(x)}_{\text{Remainder}}$$

3. Add remainder $r(x)$ to $x^{n-k}i(x)$ (puts check bits in the n-k low-order positions).

$$b(x) = x^{n-k}i(x) + r(x) \longleftarrow \text{Transmitted codeword}$$

FIGURE 3.56 Encoding procedure

Ex. 1015.91

Generator polynomial: $g(x) = x^3 + x + 1$

Information: $(1,1,0,0) \rightarrow i(x) = x^3 + x^2$

Encoding: $x^3 i(x) = x^6 + x^5$

$$
\begin{array}{r|l}
 & x^3 + x^2 + x \\
\hline
x^3 + x + 1 & x^6 + x^5 \\
 & x^6 + \quad x^4 + x^3 \\
\hline
 & x^5 + x^4 + x^3 \\
 & x^5 + \quad x^3 + x^2 \\
\hline
 & x^4 + \quad x^2 \\
 & x^4 + \quad x^2 + x \\
\hline
 & x
\end{array}
\qquad
\begin{array}{r|l}
 & 1110 \\
\hline
1011 & 1100000 \\
 & 1011 \\
\hline
 & 1110 \\
 & 1011 \\
\hline
 & 1010 \\
 & 1011 \\
\hline
 & 010
\end{array}
$$

Transmitted codeword:

$$b(x) = x^6 + x^5 + x$$

$$\rightarrow \underline{b} = (1,1,0,0,0,1,0)$$

FIGURE 3.57 Example of CRC encoding

information bits. We thus see that this encoding process introduces a binary polynomial in which the k higher-order terms are the information bits and in which the $n - k$ lower-order terms are the cyclic redundancy check bits. In the example in Figure 3.57, the division of $x^3 i(x)$ by $g(x)$ gives the remainder polynomial $r(x) = x$. The codeword polynomial is then $x^6 + x^5 + x$, which corresponds to the binary codeword (1,1,0,0,0,1,0). Note how the first four positions contain the original four information bits and how the lower three positions contains the CRC bits.

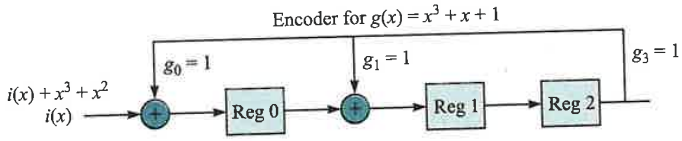
In Figure 3.55 we showed that in normal division dividing 122 by 35 yields a quotient of 3 and a remainder of 17. This result implies that $122 = 3(35) + 17$. Note that by subtracting the remainder 17 from both sides, we obtain $122 - 17 = 3(35)$ so that $122 - 17$ is evenly divisible by 35. Similarly, the codeword polynomial $b(x)$ is divisible by $g(x)$ because

$$b(x) = x^{n-k} i(x) + r(x) = g(x)q(x) + r(x) + r(x) = g(x)q(x)$$

where we have used the fact that in modulo 2 arithmetic $r(x) + r(x) = 0$. This equation implies that *all codewords are multiples of the generator polynomial $g(x)$*. This is the pattern that must be checked by the receiver. The receiver can check to see whether the pattern is satisfied by dividing the received polynomial by $g(x)$. If the remainder is nonzero, then an error has been detected.

The familiar algorithm that is taught in elementary school for carrying out "longhand" division can be used to derive a feedback shift-register circuit that implements division. The feedback taps in this circuit are determined by the coefficients of the generator polynomial. Figure 3.58 shows the division circuit for the generated polynomial $g(x) = x^3 + x + 1$. The figure also shows the states of the registers as the algorithm implements the same division that was carried out in the previous encoding example.

Ex. 1015.92



Clock	Input	Reg 0	Reg 1	Reg 2
0	-	0	0	0
1	$1 = i_3$	1	0	0
2	$1 = i_2$	1	1	0
3	$0 = i_1$	0	1	1
4	$0 = i_0$	1	1	1
5	0	1	0	1
6	0	1	0	0
7	0	**0**	**1**	**0**
	Check bits:	$r_0 = 0$	$r_1 = 1$	$r_2 = 0$

$\rightarrow r(x) = x$

FIGURE 3.58 Shift-register circuit for generated polynomial

The same division circuit that was used by the encoder can be used by the receiver to determine whether the received polynomial is a valid codeword polynomial.

3.8.5 Standardized Polynomial Codes

Table 3.8 gives generator polynomials that have been endorsed in a number of standards. The CRC-12 and CRC-16 polynomials were introduced as part of the IBM bisync protocol for controlling errors in a communication line. The CCITT-16 polynomial is used in the HDLC standard and in XMODEM. The CCITT-32 is used in IEEE 802 LAN standards and in Department of Defense protocols, as well as in the CCITT V.42 modem standard. Finally CRC-8 and CRC-10 have recently been recommended for use in ATM networks. In the problem section we explore properties and implementations of these generator polynomials.

Name	Polynomial	Used in
CRC-8	$x^8 + x^2 + x + 1$	ATM header error check
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$	ATM AAL CRC
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ $= (x+1)(x^{11} + x^2 + 1)$	Bisync
CRC-16	$x^{16} + x^{15} + x^2 + x + 1$ $= (x+1)(x^{15} + x + 1)$	Bisync
CCITT-16	$x^{16} + x^{12} + x^5 + 1$	HDLC, XMODEM, V.41
CCITT-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10}$ $+ x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	IEEE 802, DoD, V.42, AAL5

TABLE 3.8 Standard generator polynomials

Ex. 1015.93

3.8.6 Error-Detecting Capability of a Polynomial Code

We now determine the set of channel errors that a polynomial code cannot detect. In Figure 3.59 we show an additive error model for the polynomial codes. The channel can be viewed as adding, in modulo 2 arithmetic, an error polynomial, which has 1s where errors occur, to the input codeword to produce the received polynomial $R(x)$:

$$R(x) = b(x) + e(x).$$

At the receiver, $R(x)$ is divided by $g(x)$ to obtain the remainder that is defined as the **syndrome polynomial** $s(x)$. If $s(x) = 0$, then $R(x)$ is a valid codeword and is delivered to the user. If $s(x) \neq 0$, then an alarm is set, alerting the user to the detected error. Because

$$R(x) = b(x) + e(x) = g(x)q(x) + e(x)$$

we see that if an error polynomial $e(x)$ is divisible by $g(x)$, then the error pattern will be undetectable.

The design of a polynomial code for error detection involves first identifying the error polynomials we want to be able to detect and then synthesizing a generator polynomial $g(x)$ that will not divide the given error polynomials. Figures 3.60 and Figure 3.61 show the conditions required of $g(x)$ to detect various classes of error polynomials.

First consider single errors. The error polynomial is then of the form $e(x) = x^i$. Because $g(x)$ has at least two nonzero terms, it is easily shown that when multiplied by any quotient polynomial the product will also have at least two nonzero terms. Thus single errors cannot be expressed as a multiple of $g(x)$, and hence all single errors are detectable.

An error polynomial that has double errors will have the form $e(x) = x^i + x^j = x^i(1 + x^{j-i})$ where $j > i$. From the discussion for single errors, $g(x)$ cannot divide x^i. Thus $e(x)$ will be divisible by $g(x)$ only if $g(x)$ divides $(1 + x^{j-i})$. Since i can assume values from 0 to $n - 2$, we are interested in having $1 + x^m$ not be divisible by $g(x)$ for m assuming values from 1 to the maximum possible codeword length for which the polynomial will be used. The class of primitive polynomials has the property that if a polynomial has degree N, then the smallest value of m for which $1 + x^m$ is divisible by the polynomial is $2^N - 1$. [Lin 1983]. Thus if $g(x)$ is selected to be a primitive polynomial with degree $N = n - k$, then it will detect all double errors as long as the total codeword length does not exceed $2^{n-k} - 1$. Several of the generator polynomials used in practice are of the form $g(x) = (1 + x)p(x)$ where $p(x)$ is a primitive polynomial. For example, the CRC-16 polynomial is $g(x) = (1 + x)(x^{15} + x + 1)$ where $p(x) = x^{15} + x + 1$ is a

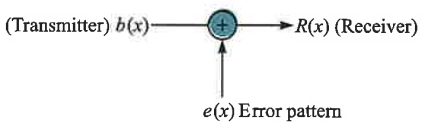


FIGURE 3.59 Additive error model for polynomial codes

1. Single errors: $e(x) = x^i \quad 0 \le i \le n\text{-}1$
If $g(x)$ has more than one term, it cannot divide $e(x)$.

2. Double errors: $e(x) = x^i + x^j \quad 0 \le i \le j \le n\text{-}1$
$= x^i(1 + x^{j-i})$
If $g(x)$ is primitive, it will not divide $(1 + x^{j-i})$ for $j\text{-}i \le 2^{n-k} - 1$.

3. Odd number of errors: $e(1) = 1$ if number of errors is odd.
If $g(x)$ has $(x + 1)$ as a factor, then $g(1) = 0$ and all codewords have an even number of 1s.

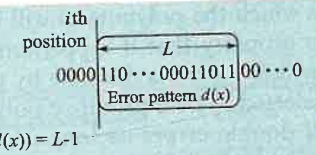
FIGURE 3.60 Generator polynomials for detecting errors—part 1

primitive polynomial. Thus this $g(x)$ will detect all double errors as long as the codeword length does not exceed $2^{15} - 1 = 32{,}767$.

Now suppose that we are interested in being able to detect all odd numbers of errors. If we can ensure that all code polynomials have an even number of 1s, then we will achieve this error-detection capability. If we evaluate the codeword polynomial $b(x)$ at $x = 1$, we then obtain the sum of the binary coefficients of $b(x)$. If $b(1) = 0$ for all codeword polynomials, then $x + 1$ must be a factor of all $b(x)$ and hence $g(x)$ must contain $x + 1$ as a factor. For this reason $g(x)$ is usually chosen so that it has $x + 1$ as a factor.

Finally consider the detection of a burst of errors of length L. As shown in Figure 3.61, the error polynomial has the form $x^i d(x)$. If the error burst involves d consecutive bits, then the degree of $d(x)$ is $L - 1$. Reasoning as before, $e(x)$ will be a multiple of $g(x)$ only if $d(x)$ is divisible by $g(x)$. Now if the degree of $d(x)$ is less than that of $g(x)$, then it will not be possible to divide $d(x)$ by $g(x)$. We conclude that if $g(x)$ has degree $n - k$, then all bursts of length $n - k$ or less will be detected.

4. Error bursts of length b:



$e(x) = x^i d(x)$ where $\deg(d(x)) = L\text{-}1$
$g(x)$ has degree $n\text{-}k$;
$g(x)$ cannot divide $d(x)$ if $\deg(g(x)) > \deg(d(x))$

- $L = (n\text{-}k)$ or less: all will be detected
- $L = (n\text{-}k + 1)$: $\deg(d(x)) = \deg(g(x))$
  i.e. $d(x) = g(x)$ is the only undetectable error pattern,
  fraction of bursts that are undetectable $= 1/2^{L-2}$
- $L > (n\text{-}k + 1)$: fraction of bursts that are undetectable $= 1/2^{n-k}$

FIGURE 3.61 Generator polynomials for detecting errors—part 2

If the burst error has length $L = n - k + 1$, that is, degree of $d(x) =$ degree of $g(x)$, then $d(x)$ is divisible by $g(x)$ only if $d(x) = g(x)$. From Figure 3.61 $d(x)$ must have 1 in its lowest-order term and in its highest-order term, so it matches $g(x)$ in these two coefficients. For $d(x)$ to equal $g(x)$, it must also match $g(x)$ in the $n - k - 1$ coefficients that are between the lowest- and highest-order terms. Only one of the 2^{n-k-1} such patterns will match $g(x)$. Therefore, the proportion of bursts of length $L = n - k + 1$ that is undetectable is $1/2^{n-k-1}$. Finally, it can be shown that in the case of $L > n - k + 1$ the fraction of bursts that is undetectable is $1/2^{n-k}$.

◆3.8.7 Linear Codes[13]

We now introduce the class of linear codes that are used extensively for error detection and correction. A **binary linear code** is specified by two parameters: k and n. The linear code takes groups of k information bits, $b_1, b_2, \ldots, b_k$, and produces a binary codeword $\underline{b}$ that consists of n bits, $b_1, b_2, \ldots, b_n$. The $n - k$ check bits $b_{k+1}, \ldots, b_n$, are determined by $n - k$ linear equations:[14]

$$\begin{aligned} b_{k+1} &= a_{11}b_1 + a_{12}b_2 + \ldots + a_{1k}b_k \\ b_{k+2} &= a_{21}b_1 + a_{22}b_2 + \ldots + a_{2k}b_k \\ &\vdots \\ b_n &= a_{(n-k)1}b_1 + a_{(n-k)2}b_2 + \ldots + a_{(n-k)k}b_k \end{aligned} \tag{5}$$

The coefficients in the preceding equations are binary numbers, and the addition is modulo 2. We say that b_{k-j} checks the information bit b_i if a_{ji} is 1. Therefore b_{k-j} is given by the modulo 2 sum of the information bits that it checks, and thus the redundancy in general linear codes is determined by parity check sums on subsets of the information bits. Note that when all of the information bits are 0, then all of the check bits will be 0. Thus the n-tuple $\underline{0}$ with all zeros is always one of the codewords of a linear code. Many linear codes can be defined by selecting different coefficients $[a_{ji}]$. Coding books contain catalogs of good codes that can be selected for various applications.

In linear codes the redundancy is provided by the $n - k$ check bits. Thus if the transmission channel has a bit rate of R bits/seconds, then k of every n transmitted bits are information bits, so the rate at which user information flows through the channel is $R_{\text{info}} = (k/n)R$ bits/second.

As an example consider the (7,4) linear **Hamming code** in which the first four bits of the codeword $\underline{b}$ consist of the four information bits b_1, b_2, b_3, and b_4 and the three check bits b_5, b_6, and b_7 are given by

[13]Section titles preceded by ◆ provide additional details and are not essential for subsequent sections.
[14]We require the set of linear equations to be linearly independent; that is, no equation can be written as a linear combination of the other equations.

Ex. 1015.96

$$
\begin{aligned}
b_5 &= b_1 \qquad\quad + b_3 + b_4 \\
b_6 &= b_1 + b_2 \qquad\quad + b_4 \\
b_7 &= \qquad + b_2 + b_3 + b_4
\end{aligned}
\tag{6}
$$

We have arranged the preceding equations so that it is clear which information bits are being checked by which check bits, that is, b_5 checks information bits b_1, b_3, and b_4. These equations allow us to determine the codeword for any block of information bits. For example, if the four information bits are (0,1,1,0), then the codeword is given by $(0, 1, 1, 0, 0+1+0, 0+1+0, 0+1+0, 1+1+0) = (0, 1, 1, 0, 1, 1, 0)$. Table 3.9 shows the set of 16 codewords that are assigned to the 16 possible information blocks.

Linear codes provide a very simple method for detecting errors. Before considering the general case, we illustrate the method using the Hamming code (Table 9). Suppose that in equation 6 we add b_5 to both sides of the first equation, b_6 to both sides of the second equation, and b_7 to both sides of the third equation. We then obtain

$$
\begin{aligned}
0 = b_5 + b_5 &= b_1 \qquad\quad + b_3 + b_4 + b_5 \\
0 = b_6 + b_6 &= b_1 + b_2 \qquad\quad + b_4 \qquad\quad + b_6 \\
0 = b_7 + b_7 &= \qquad + b_2 + b_3 + b_4 \qquad\qquad\qquad + b_7
\end{aligned}
$$

where we have used the fact that in modulo 2 arithmetic any number plus itself is always zero. The preceding equations state the conditions that must be satisfied by every codeword. Thus if $\underline{r} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7)$ is the output of the transmission channel, then $\underline{r}$ is a codeword only if its components satisfy these equa-

Information				Codeword							Weight
b_1	b_2	b_3	b_4	b_1	b_2	b_3	b_4	b_5	b_6	b_7	$w(\underline{b})$
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	4
0	0	1	0	0	0	1	0	1	0	1	3
0	0	1	1	0	0	1	1	0	1	0	3
0	1	0	0	0	1	0	0	0	1	1	3
0	1	0	1	0	1	0	1	1	0	0	3
0	1	1	0	0	1	1	0	1	1	0	4
0	1	1	1	0	1	1	1	0	0	1	4
1	0	0	0	1	0	0	0	1	1	0	3
1	0	0	1	1	0	0	1	0	0	1	3
1	0	1	0	1	0	1	0	0	1	1	4
1	0	1	1	1	0	1	1	1	0	0	4
1	1	0	0	1	1	0	0	1	0	1	4
1	1	0	1	1	1	0	1	0	1	0	4
1	1	1	0	1	1	1	0	0	0	0	3
1	1	1	1	1	1	1	1	1	1	1	7

TABLE 3.9 Hamming (7,4) code

Ex. 1015.97

tions. If we write the equations in matrix form, we obtain a more compact representation for the conditions that all codewords must meet.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1011100 \\ 1101010 \\ 0111001 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \mathbf{H}\underline{b}^t = \underline{0}$$

In the matrix **H** is defined as the 3×7 matrix and $\underline{b}^t$ is the codeword arranged as a column vector. (Above we had previously defined $\underline{b}$ as row vector.)

By following the same procedure we have used in this example, we find that *the condition that all codewords must meet* is given by the matrix equation:

$$\mathbf{H}\underline{b}' = \underline{0} \tag{7}$$

where $\underline{0}$ is a column vector with $n-k$ components all equal to zero, $\underline{b}^t$ is the codeword arranged as a column vector, and **check matrix H** has dimension $n-k$ rows by n columns and is given by

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & 10\ldots0 \\ a_{21} & a_{22} & \cdots & a_{2k} & 01\ldots0 \\ & & \vdots & & \ddots \\ a_{(n-k)1} & a_{(n-k)2} & \cdots & a_{(n-k)k} & 00\ldots1 \end{bmatrix}$$

Errors introduced in a binary channel can be modeled by the additive process shown in Figure 3.62a. The output of the binary channel can be viewed as the modulo 2 addition of an error bit e to the binary input b. If the error bit equals 0, then the output of the channel will correspond to the input of the channel and no transmission error occurs; if the error bit equals 1, then the output will differ from the input and a transmission error occurs. Now consider the effect of the binary channel on a transmitted codeword. As shown in Figure 3.62b, the

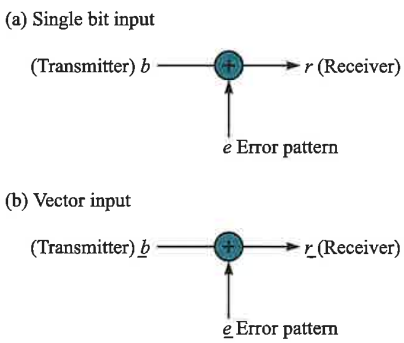


FIGURE 3.62 Additive error channel

channel can be viewed as having a vector input that consists of the n bits that correspond to the codeword. The output of the channel $\underline{r}$ is given by the modulo 2 sum of the codeword $\underline{b}$ and an error vector $\underline{e}$ that has 1s in the components where an error occurs and 0s elsewhere:

$$\underline{r} = \underline{b} + \underline{e}$$

The error-detection system that uses a linear code checks the output of a binary channel $\underline{r}$ to see whether $\underline{r}$ is a valid codeword. The system does this by checking to see whether $\underline{r}$ satisfies equation 7. The result of this calculation is an $(n-k) \times 1$ column vector called the **syndrome**:

$$\underline{s} = H\underline{r} \tag{8}$$

If $\underline{s} = \underline{0}$, then $\underline{r}$ is a valid codeword; therefore, the system assumes that no errors have occurred and delivers $\underline{r}$ to the user. If $\underline{s} \neq \underline{0}$, then $\underline{r}$ is not a valid codeword and the error-detection system sets an alarm indicating that errors have occurred in transmission. In an ARQ system a retransmission is requested in response to the alarm. In an FEC system the alarm would initiate a processing based on the syndrome that would attempt to identify which bits were in error and then proceed to correct them.

The error-detection system fails when $\underline{s} = 0$ but the output of the channel is not equal to the input of the channel; that is, $\underline{e}$ is nonzero. In terms of equation 8 we have

$$\underline{0} = \underline{s} = H\underline{r} = H(\underline{b} + \underline{e}) = H\underline{b} + H\underline{e} = \underline{0} + H\underline{e} = H\underline{e} \tag{9}$$

where the fourth equality results from the linearity property of matrix multiplication and the fifth equality uses equation 7. The equality $H\underline{e} = \underline{0}$ implies that when $\underline{s} = \underline{0}$ the error pattern $\underline{e}$ satisfies equation 7 and hence must be a codeword. This implies that error detection using linear codes fails when the error vector is a codeword that transforms the input codeword $\underline{b}$ into a different codeword $\underline{r} = \underline{b} + \underline{e}$. Thus the set of all undetectable error vectors is the set of all nonzero codewords, and the probability of detection failure is the probability that the error vector equals any of the nonzero codewords.

In Figure 3.63 we show an example of the syndrome calculation using the (7,4) Hamming code for error vectors that contain single, double, and triple errors. We see from the example that if a single error occurred in the jth position, then the syndrome will be equal to the jth column of the **H** matrix. Since all the columns of **H** are nonzero, it follows that the syndrome will always be nonzero when the error vector contains a single error. Thus all single errors are detectable. The second example shows that if the error vector contains an error in location i and an error in location j, then the syndrome is equal to the sum of the ith and jth columns of **H**. We note that for the Hamming (7,4) code all columns are distinct. Thus the syndrome will be nonzero, and all error vectors with two errors are detectable. The third example shows an error vector that contains three errors. The syndrome for this particular error vector is zero. Thus we find that this Hamming code can detect all single and double errors but fails to detect some triple errors.

Ex. 1015.99

$$\underline{s} = \mathbf{H}\underline{e} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \text{Single error detected}$$

$$\underline{s} = \mathbf{H}\underline{e} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \text{Double error detected}$$

$$\underline{s} = \mathbf{H}\underline{e} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \underline{0} \qquad \text{Triple error not detected}$$

FIGURE 3.63 Syndrome calculation

A general class of Hamming codes can be defined with **H** matrices that satisfy the properties identified in the preceding example. Note that in the Hamming (7,4) code each of the $2^3 - 1$ possible nonzero binary triplets appears once and only once as a column of the H matrix. This condition enables the code to detect all single and double errors. Let m be an integer greater than or equal to 2. We can then construct an **H** matrix that has as its columns the $2^m - 1$ possible nonzero binary m-tuples. This **H** matrix corresponds to a linear code with codewords of length $n = 2^m - 1$ and with $n - k = m$ check bits. The codes that have this **H** matrix are called Hamming codes, and they are all capable of detecting all error vectors that have single and double errors. In the examples, we have been using the $m = 3$ Hamming code.

PERFORMANCE OF LINEAR CODES

In Figure 3.51 we showed qualitatively that we can minimize the probability of error-detection failure by spacing codewords apart in the sense that it is unlikely for errors to convert one codeword into another. In this section we show that the error-detection performance of a code is determined by the distances between codewords.

The **Hamming distance** $d(\underline{b}_1, \underline{b}_2)$ between the binary vectors $\underline{b}_1$ and $\underline{b}_2$ is defined as the number of components in which they differ. Thus the Hamming distance between two vectors increases as the number of bits in which they differ increases. Consider the modulo 2 sum of two binary n-tuples $\underline{b}_1 + \underline{b}_2$. The components of this sum will equal one when the corresponding components in $\underline{b}_1$ and $\underline{b}_2$ differ, and they will be zero otherwise. Clearly, this result is equal to the number of 1s in $\underline{b}_1 + b_2$, so

$$d(\underline{b}_1, \underline{b}_2) = w(\underline{b}_1 + \underline{b}_2) \tag{10}$$

where w is the weight function introduced earlier. The extent to which error vectors with few errors are more likely than error vectors with many errors suggests that we should design linear codes that have codewords that are far apart in the sense of Hamming distance.

Define the **minimum distance** d_{min} of a code as follows:

$$d_{min} = \textit{distance between two closest distinct codewords}$$

For any given linear code, the pair of closest codewords is the most vulnerable to transmission error, so d_{min} can be used as a worst-case type of measure. From equation 9 we have that if $\underline{b}_1$ and $\underline{b}_2$ are codewords, then $\underline{b}_1 + \underline{b}_2$ is also a codeword. To find d_{min}, we need to find the pair of distinct codewords $\underline{b}_1$ and $\underline{b}_2$ that minimize $d(\underline{b}_1, \underline{b}_2)$. By equation 10, this is equivalent to finding the nonzero codeword with the smallest weight. Thus

$$d_{min} = \textit{weight of the nonzero codeword with the smallest number of } 1s$$

From Table 3.9 in section 3.8.7, we see the Hamming (7,4) code has $d_{min} = 3$.

If we start changing the bits in a codeword one at a time until another codeword is obtained, then we will need to change at least d_{min} bits before we obtain another codeword. This situation implies that all error vectors with $d_{min} - 1$ or fewer errors are detectable. We say that **a code is t-error detecting** if $d_{min} \geq t + 1$.

Finally, let us consider the probability of error-detection failure for a general linear code. In the case of the random error vector channel model, all 2^n possible error patterns are equally probable. A linear (n, k) code fails to detect only the $2^k - 1$ error vectors that correspond to nonzero codewords. We can state then that *the probability of error-detection failure for the random error vector channel model is* $(2^k - 1)/2^n \approx 1/2^{n-k}$. Furthermore, we can decrease the probability of detection failure by increasing the number of parity bits $n - k$.

Consider now the random bit error channel model. The probability of detection failure is given by

$$
\begin{aligned}
P[\text{detection failure}] &= P[\underline{e} \text{ is a nonzero codeword}] \\
&= \sum_{\text{nonzero codewords } \underline{b}} (1-p)^{n-w(\underline{b})} p^{w(\underline{b})} \\
&= \sum_{w=d_{\min}}^{d_{\max}} N_w (1-p)^{n-w} p^w \\
&\approx N_{d_{\min}} p^{d_{\min}} \qquad \text{for } p \ll 1
\end{aligned}
$$

The second summation adds the probability of all nonzero codewords. The third summation combines all codewords of the same weight, so N_w is the total number of codewords that have weight w. The approximation results from the fact that the summation is dominated by the leading term when p is very small.

Consider the (7,4) Hamming code as an example once again. For the random error vector model, the probability of error-detection failure is $1/2^3 = 1/8$. On the other hand, for the random bit error channel the probability of error-detection failure is approximately $7p^3$, since $d_{min} = 3$ and seven codewords have this weight. If $p = 10^{-4}$, then the probability of error detection failure is 7×10^{-12}. Compared to the single parity check code, the Hamming code yields a tremendous improvement in error-detection capability.

◆3.8.8 Error Correction

In FEC the detection of transmission errors is followed by processing to determine the most likely error locations. Assume that an error has been detected so $\underline{s} \neq \underline{0}$. Equation 11 describes how an FEC system attempts to carry out the correction.

$$\underline{s} = H\underline{r} = H(\underline{b} + \underline{e}) = H\underline{b} + H\underline{e} = \underline{0} + H\underline{e} = H\underline{e}. \tag{11}$$

The receiver uses equation 11 to find the value of the syndrome to diagnose the most likely error pattern. If $\mathbf{H}$ were an invertible matrix, then we could readily find the error vector from $\underline{e} = \mathbf{H}^{-1}\underline{s}$. Unfortunately, $\mathbf{H}$ is not invertible. Equation 11 consists of $n-k$ equations in n unknowns, $e_1, e_2, \ldots, e_n$. Because we have fewer equations than unknowns, the system is underdetermined and equation 11 has more than one solution. In fact, it can be shown that 2^k binary n-tuples satisfy equation 11. Thus for any given nonzero $\underline{s}$, equation 11 allows us to identify the 2^k possible error vectors that could have produced $\underline{s}$. The error-correction system cannot proceed unless it has information about the probabilities with which different error patterns can occur. The error-correction system uses such information to identify the most likely error pattern from the set of possible error patterns.

We provide a simple example to show how error correction is carried out. Suppose we are using the Hamming (7,4) code. Assume that the received vector is $\underline{r} = (0,0,1,0,0,0,1)$. The syndrome calculation gives $\underline{s} = (1,0,0)^k$. Because the fifth column of $\mathbf{H}$ is (1,0,0), one of the error vectors that gives this syndrome is

(0,0,0,0,1,0,0). Note from equation 11 that if we add a codeword to this error vector, we obtain another vector that gives the syndrome $(1,0,0)^t$. The $2^k = 16$ possible error vectors are obtained by adding the 16 codewords to (0,0,0,0,1,0,0) and are listed in Table 3.10. The error-correction system must now select the error vector in this set that is most likely to have been introduced by the channel. Almost all error-correction systems simply select the error vector with the smallest number of 1s. Note that this error vector also corresponds to the most likely error vector for the random bit error channel model. For this example the error correction system selects $\underline{e} = (0,0,0,0,1,0,0)$ and then outputs the codeword $\underline{r} + \underline{e} = (0,0,1,0,1,0,1)$ from which the user extracts the information bits, 0010. Algorithms have been developed that allow the calculation of the most likely error vector from the syndrome. Alternatively, the calculations can be carried out once, and then a table can be set up that contains the error vector that is to be used for correction for each possible syndrome. The error-correction system then carries out a table lookup each time a nonzero syndrome is found.

The error-correction system is forced to select only one error vector out of the 2^k possible error vectors that could have produced the given syndrome. Thus the error-correction system will successfully recover the transmitted codeword only if the error vector is the most likely error vector in the set. When the error vector is one of the other 2^k possible error vectors, the error-correction system will perform corrections in the wrong locations and actually introduce more errors! In the preceding example, assuming a random bit error channel model, the probability of the most likely error vector is $p(1-p)^6 \approx p$ for the error vector of weight 1; the probability of the other error vectors is approximately $3p^2(1-p)^5$ for the three error vectors of weight 2 and for where we have neglected the remainder of the error patterns. Thus when the error-correction system detects

Error vectors							Weight
e_1	e_2	e_3	e_4	e_5	e_6	e_7	$w(\underline{e})$
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	3
0	0	1	0	0	0	1	2
0	0	1	1	1	1	0	4
0	1	0	0	1	1	1	4
0	1	0	1	0	0	0	2
0	1	1	0	0	1	0	3
0	1	1	1	1	0	1	5
1	0	0	0	0	1	0	2
1	0	0	1	1	0	1	4
1	0	1	0	1	1	1	5
1	0	1	1	0	0	0	3
1	1	0	0	0	0	1	3
1	1	0	1	1	1	0	5
1	1	1	0	1	0	0	4
1	1	1	1	0	1	1	6

TABLE 3.10 Error vectors corresponding to syndrome $(\mathbf{1}, \mathbf{0}, \mathbf{0})^t$

an error the system's attempt to correct the error fails with probability

$$\frac{3p^2(1-p)^5}{p(1-p)^6 + 3p^2(1-p)^5} \approx 3p$$

Figure 3.64 summarizes the four outcomes that can arise from the error-correction process. We begin with the error vector that is revealed through its syndrome. If the $\underline{s} = \underline{0}$, then the received vector $\underline{r}$ is accepted as correct and delivered to the user. Two outcomes lead to $\underline{s} = \underline{0}$: the first corresponds to when no errors occur in transmission and has probability $(1-p)^7 \approx 1 - 7p$; the second corresponds to when the error vector is undetectable and has probability $7p^3$. If $\underline{s} \neq \underline{0}$, the system attempts to perform error correction. This situation occurs with probability $1 - P[\underline{s} = \underline{0}] = 1 - \{1 - 7p + 7p^3\} \approx 7p$. Two further outcomes are possible in the $\underline{s} \neq \underline{0}$ case: the third outcome is when the error vector is correctable and has conditional probability $(1 - 3p)$; the fourth is when the error vector is not correctable and has conditional probability $3p$. From the figure we see that the probability that the error-correction system fails to correct an error pattern is $21p^2$. To summarize, the first and third outcomes yield correct user information. The second and fourth outcomes result in the delivery of incorrect information to the user. Through this example we have demonstrated the analysis required to determine the effectiveness of any error-correction system.

The minimum distance of a code is useful in specifying its error-correcting capability. In Figure 3.65 we consider a code with $d_{min} = 5$, and we show two codewords that are separated by the minimum distance. If we start by changing the bits in $\underline{b}_1$, one bit at a time until we obtain $\underline{b}_2$, we find four n-tuples between the two codewords. We can imagine drawing a sphere of radius 2 around each codeword. The sphere around $\underline{b}_1$ will contain two of the n-tuples, and the sphere around $\underline{b}_2$ will contain the other n-tuples.

Note that because all pairs of codewords are separated by at least distance d_{min}, we can draw a sphere of radius 2 around every single codeword, and these spheres will all be nonoverlapping. This geometrical view gives us another way of looking at error correction. We can imagine that the error-correction system takes the vector $\underline{r}$ and looks up which sphere it belongs to; the system then

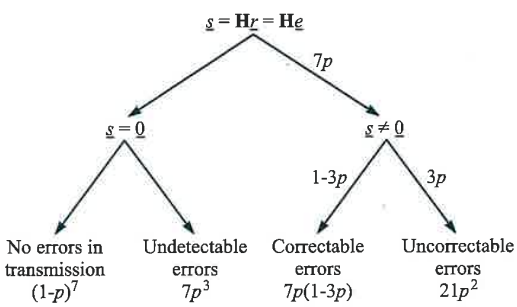
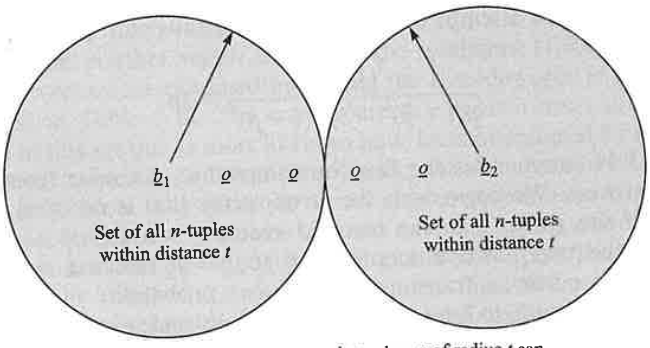


FIGURE 3.64 Summary of error-correction process outcomes



If $d_{min} = 2t + 1$, nonoverlapping spheres of radius t can be drawn around each codeword; $t = 2$ in the figure

FIGURE 3.65 Partitioning of n-tuples into disjoint spheres

generates the codeword that is at the center of the sphere. Note that if the error vector introduced by the channel has two or fewer errors, then the error-correction system will always produce the correct codeword. Conversely, if the number of errors is more than two, the error-correction system will produce an incorrect codeword.

The discussion of Figure 3.65 can be generalized as follows. Given a linear code with $d_{min} \geq 2t + 1$, it is possible to draw nonoverlapping spheres of radius t around all the codewords. Hence the error-correction system is guaranteed to operate correctly whenever the number of errors is smaller than t. For this reason we say that **a code is t-error correcting** if $d_{min} \geq 2t + 1$.

The Hamming codes introduced above all have $d_{min} = 3$. Consequently, they are all single-error correcting. The Hamming codes use $m = n - k$ bits of redundancy and are capable of correcting single errors. An interesting question is, If we use $n - k = 2m$ bits in a code of length $n = 2^m - 1$, can we correct all double errors? Similarly, if we use $n - k = 3m$, can we correct triple errors? The answer is yes in some cases and leads to the classes of BCH and Reed-Solomon codes [Lin 1983].

In this book we have presented only linear codes that operate on nonoverlapping blocks of information. These block codes include the classes of Hamming codes, BCH codes, and Reed-Solomon codes that have been studied extensively and are in wide use. These codes provide a range of choice in terms of n, k, and d_{min} that allows a system designer to select a code for a given application. Convolutional codes are another important class of error-correcting codes. These codes operate on overlapping blocks of information and are also in wide use.

Finally, we consider the problem of error correction in channels that introduce bursts of errors. The codes discussed up to this point correct error vectors that contain $(d_{min} - 1)/2$ or fewer errors. These codes can also be used in channels with burst errors if combined with **interleaving**. The user information is

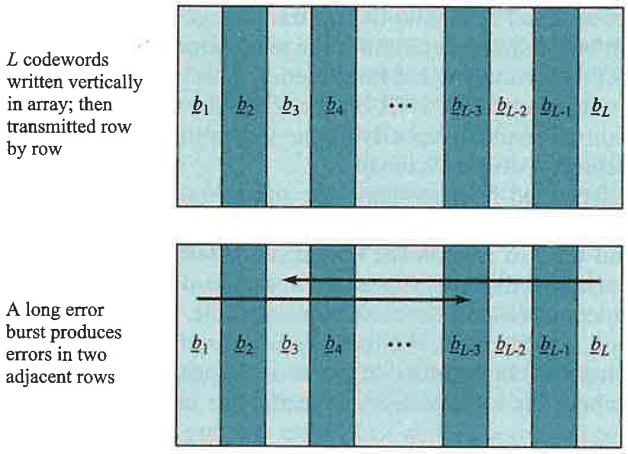


FIGURE 3.66 Interleaving

encoded using the given linear code, and the codewords are written as columns in an array as shown in Figure 3.66. The array is transmitted over the communication channels *row by row*. The interleaver depth L is selected so that the errors associated with a burst are distributed over many codewords. The error-correction system will be effective if the number of errors in each codeword is within its error-correcting capability. For example, if the linear code can correct up to two errors, then interleaving makes it possible to correct any burst of length less than $2L$.

SUMMARY

Binary information, "bits," are at the heart of modern communications. All information can be represented as blocks or streams of bits. Modern communication networks are designed to carry bits and therefore can handle *any* type of information.

We began this chapter with a discussion of the basic properties of common types of information such as text, image, voice, audio, and video. We discussed the amount of information that is required to represent them in terms of bits or bits/second. We also discussed requirements that applications impose on the network when they involve these types of information.

We described the difference between digital and analog communication and explained why digital communication has prevailed. We then considered the design of digital transmission systems. The characterization of communication channels in terms of their response to sinusoidal signals and to pulse signals was introduced. The notion of bandwidth of a channel was also introduced.

We first considered baseband digital transmission systems. We showed how the bandwidth of a channel determines the maximum rate at which pulses can be transmitted with zero intersymbol interference. This is the Nyquist signaling rate. We then showed the effect of SNR on the reliability of transmissions and developed the notion of channel capacity as the maximum reliable transmission rate that can be achieved over a channel.

Next we explained how modems use sinusoidal signals to transmit binary information over bandpass channels. The notion of a signal constellation was introduced and used to explain the operation of telephone modem standards.

The properties of different types of transmission media were discussed next. We first considered twisted-pair cable, coaxial cable, and optical fiber, which are used in "wired" transmission. We then discussed radio and infrared light, which are used in wireless transmission. Important physical layer standards were used as examples where the various types of media are used.

Finally, we presented coding techniques that are used in error control. Basic error-detection schemes that are used in many network standards were introduced first. An optional section then discussed error-correction schemes that are used when a return channel is not available.

CHECKLIST OF IMPORTANT TERMS

amplitude-response
amplitude shift keying (ASK)
asymmetric digital subscriber line (ADSL)
attenuation
automatic retransmission request (ARQ)
bandwidth
baseband transmission
◆binary linear code
bipolar encoding
bit error rate
bit rate
block-oriented information
burst error
cable modem
channel
channel capacity
check bit
◆check matrix
checksum
coaxial cable
codeword
compression ratio
cyclic redundancy check (CRC)
data compression
delay
differential encoding
digital information
digital transmission
equalizer
error control
forward error correction (FEC)
frequency shift keying (FSK)
generator polynomial
◆Hamming code
◆Hamming distance
impulse response
information polynomial
in-phase component
◆interleaving
jitter
line coding
loading coils

loss
lossless data compression
lossy data compression
Manchester encoding
◆minimum distance
modem
multilevel transmission
multimode fiber
nonreturn-to-zero (NRZ) encoding
NRZ inverted
Nyquist signaling rate
optical fiber
path loss exponent
phase shift
phase shift keying (PSK)
physical transmission medium
polynomial code
pulse code modulation (PCM)
quadrature amplitude modulation (QAM)
quadrature-phase component
random bit error channel model
random error vector model
redundancy
signal constellation
signal-to-noise ratio (SNR)
single parity check code
single-mode fiber
◆syndrome
syndrome polynomial
◆t-error detecting
transmission error
transmission medium
trellis modulation
twisted pair
unshielded twisted pair (UTP)
wavelength
wavelength division multiplexing (WDM)
weight

FURTHER READING

Ahamed, S. W. and V. B. Lawrence, *Design and Engineering of Intelligent Communication Systems*, Kluwer Academic Publishers, Boston, 1997. Detailed information about properties of transmission media.

Bell Telephone Laboratories, *Transmission Systems for Communications*, 1971. Classic on transmission aspects of telephony.

Bellamy, J., *Digital Telephony*, John Wiley & Sons, Inc., New York, 1991. Excellent coverage of digital telephony.

Dutton, H. J. R. and P. Lenhard, *High-Speed Networking Technology*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1995. Good coverage of physical layer aspects of computer networks.

Gigabit Ethernet Alliance, "Accelerating the Standard for Speed," White paper, http://www.gigabit-ethernet.org, 1998.

Glover, I. A. and P. M. Grant, *Digital Communications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1998. Up-to-date introduction to digital transmission systems.

Leon-Garcia, A., *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, Reading, Massachusetts, 1994.

Lin, S. and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1983.

Mukherjee, B., *Optical Communication Networks*, McGraw-Hill, New York, 1997.

Seifert, R., *Gigabit Ethernet*, Addison-Wesley, Reading, Massachusetts, 1998.

Smith, D. R., *Digital Transmission Systems*, Van Nostrand Reinhold Company, New York, 1985.

RFC 1071, R. Braden and D. Dorman, "Computing the Internet Checksum," September 1988.

PROBLEMS

1. Suppose the size of an uncompressed text file is 1 megabyte.
   a. How long does it take to download the file over a 32 kilobit/second modem?
   b. How long does it take to download the file over a 1 megabit/second modem?
   c. Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

2. A scanner has a resolution of 600 × 600 pixels/square inch. How many bits are produced by an 8-inch-×-10-inch image if scanning uses 8 bits/pixel? 24 bits/pixel?

3. Suppose a computer monitor has a screen resolution of 1200 × 800 pixels. How many bits are required if each pixel uses 256 colors? 65,536 colors?

4. Explain the difference between facsimile, GIF, and JPEG coding. Give an example of an image that is appropriate to each of these three methods.

5. A digital transmission system has a bit rate of 45 Megabits/second. How many PCM voice calls can be carried by the system?

6. Suppose a storage device has a capacity of 1 gigabyte. How many 1-minute songs can the device hold using conventional CD format? using MP3 coding?

7. How many high-quality audio channels can be transmitted using an HDTV channel?

8. How many HDTV channels can be transmitted simultaneously over the optical fiber transmission systems in Table 3.3?

9. Comment on the properties of the sequence of frame images and the associated bit rates in the following examples:
   a. A children's cartoon program.
   b. A music video.
   c. A tennis game; a basketball game.
   d. A documentary on famous paintings.

10. Suppose that at a given time of the day, in a city with a population of 1 million, 1% of the people are on the phone.
    a. What is the total bit rate generated by all these people if each voice call is encoded using PCM?
    b. What is the total bit rate if all of the telephones are replaced by H.261 videoconferencing terminals?

11. Consider an analog repeater system in which the signal has power σ_x^2 and each stage adds noise with power σ_n^2. For simplicity assume that each repeater recovers the original signal without distortion but that the noise accumulates. Find the SNR after n repeater links. Write the expression in decibels: SNR dB $=$ 10 $\log_{10}$SNR.

12. Suppose that a link between two telephone offices has 50 repeaters. Suppose that the probability that a repeater fails during a year is 0.01 and that repeaters fail independently of each other.

a. What is the probability that the link does not fail at all during one year?

b. Repeat (a) with 10 repeaters; with 1 repeater.

13. Suppose that a signal has twice the power as a noise signal that is added to it. Find the SNR in decibels. Repeat if the signal has 10 times the noise power? 2^n times the noise power? 10^k times the noise power?

14. A square periodic signal is represented as the following sum of sinusoids:

$$g(t) = \frac{2}{\pi}\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\cos(2k+1)\pi t$$

a. Suppose that the signal is applied to an ideal low-pass filter with bandwidth 15 Hz. Plot the output from the low-pass filter and compare to the original signal. Repeat for 5 Hz; for 3 Hz. What happens as W increases?

b. Suppose that the signal is applied to a bandpass filter that passes frequencies from 5 to 9 Hz. Plot the output from the filter and compare to the original signal.

15. Suppose that the 8 kbps periodic signal in Figure 3.15 is transmitted over a system that has an attenuation function equal to one for all frequencies and a phase function that is equal to $-90°$ for all frequencies. Plot the signal that comes out of this system. Does it differ in shape from the input signal?

16. A 10 kHz baseband channel is used by a digital transmission system. Ideal pulses are sent at the Nyquist rate, and the pulses can take 16 levels. What is the bit rate of the system?

17. Suppose a baseband transmission system is constrained to a maximum signal level of ± 1 volt and that the additive noise that appears in the receiver is uniformly distributed between $[-1/16, 1/16]$. How many levels of pulses can this transmission system use before the noise starts introducing errors?

18. What is the maximum reliable bit rate possible over a telephone channel with the following parameters:

a. $W = 2.4$ kHz SNR $=$ 20 dB

b. $W = 2.4$ kHz SNR $=$ 40 dB

c. $W = 3.0$ kHz SNR $=$ 20 dB

d. $W = 3.0$ kHz SNR $=$ 40 dB

19. Suppose we wish to transmit at a rate of 64 kbps over a 3 kHz telephone channel. What is the minimum SNR required to accomplish this?

20. Suppose that a low-pass communications system has a 1 MHz bandwidth. What bit rate is attainable using 8-level pulses? What is the Shannon capacity of this channel if the SNR is 20 dB? 40 dB?

21. Most digital transmission systems are "self-clocking" in that they derive the bit synchronization from the signal itself. To do this, the systems use the transitions between positive and negative voltage levels. These transitions help define the boundaries of the bit intervals.

a. The nonreturn-to-zero (NRZ) signaling method transmits a 0 with a $+1$ voltage of duration T, and a 1 with a -1 voltage of duration T. Plot the signal for the sequence n consecutive 1s followed by n consecutive 0s. Explain why this code has a synchronization problem.

b. In differential coding the sequence of 0s and 1s induces changes in the polarity of the signal; a binary 0 results in no change in polarity, and a binary 1 results in a change in polarity. Repeat part (a). Does this scheme have a synchronization problem?

c. The Manchester signaling method transmits a 0 as a $+1$ voltage for $T/2$ seconds followed by a -1 for $T/2$ seconds; a 1 is transmitted as a -1 voltage for $T/2$ seconds followed by a $+1$ for $T/2$ seconds. Repeat part (a) and explain how the synchronization problem has been addressed. What is the cost in bandwidth in going from NRZ to Manchester coding?

22. Consider a baseband transmission channel with a bandwidth of 10 MHz. Which bit rates can be supported by the bipolar line code and by the Manchester line code?

23. The impulse response in a T-1 copper-wire transmission system has the idealized form where the initial pulse is of amplitude 1 and duration 1 and the afterpulse is of amplitude -0.1 and of duration 10.

a. Let $\delta(t)$ be the narrow input pulse in Figure 3.18a. Suppose we use the following signaling method: Every second, the transmitter accepts an information bit; if the information bit is 0, then $-\delta(t)$ is transmitted, and if the information bit is 1, then $\delta(t)$ is transmitted. Plot the output of the channel for the sequence 1111000. Explain why the system is said to have "dc" or baseline wander.

b. The T-1 transmission system uses bipolar signaling in the following fashion: If the information bit is a 0, then the input to the system is $0^*\delta(t)$; if the information bit is a 1, then the input is $\delta(t)$ for an even occurrence of a 1 and $-\delta(t)$ for an odd occurrence of a 1. Plot the output of the channel for the sequence 1111000. Explain how this signaling solves the "dc" or baseline wander problem.

24. The raised cosine transfer function, shown in Figure 3.21, has a corresponding impulse response given by

$$p(t) = \frac{\sin(\pi t/T)}{\pi t/T}\,\frac{\cos(\pi\alpha t/T)}{1-(2\alpha t/t)^2}$$

a. Plot the response of the information sequence 1010 for $\alpha = \frac{1}{2}$; $\alpha = \frac{1}{8}$.

b. Compare this plot to the response, using the pulse in Figure 3.17.

25. Suppose a CATV system uses coaxial cable to carry 100 channels, each of 6 MHz bandwidth. Suppose that QAM modulation is used.

a. What is the bit rate/channel if a four-point constellation is used? eight-point constellation?
b. Suppose a digital TV signal requires 4 Mbps. How many digital TV signals can each channel handle for the two cases in part (a)?

26. Explain how ASK was used in radio telegraphy. Compare the use of ASK to transmit Morse code with the use of ASK to transmit text using binary information.

27. Suppose that a modem can transmit eight distinct tones at distinct frequencies. Every T seconds the modem transmits an arbitrary combination of tones (that is, some are present, and some are not present).
a. What bit rate can be transmitted using this modem?
b. Is there a relationship between T and the frequency of the signals?

28. A phase modulation system transmits the modulated signal $A\cos(2\pi f_c t + \phi)$ where the phase ϕ is determined by the two information bits that are accepted every T-second interval:
for 00 $\phi = 0$; for 01 $\phi = \pi/2$; for 10 $\phi = \pi$; for 11 $\phi = 3\pi/2$.
a. Plot the signal constellation for this modulation scheme.
b. Explain how an eight-point phase modulation scheme would operate.

29. Suppose that the receiver in a QAM system is not perfectly synchronized to the carrier of the received signal; that is, the receiver multiplies the received signal by $2\cos(2\pi f_c t + \phi)$ and by $2\sin(2\pi/f_c t + \phi)$ where ϕ is a small phase error. What is the output of the demodulator?

30. In differential phase modulation the binary information determines the *change* in the phase of the carrier signal $\cos(2\pi f_c t)$. For example, if the information bits are 00, the phase change is 0; if 01, it is $\pi/2$; for 10, it is π; and for 11, it is $3\pi/2$.
a. Plot the modulated waveform that results from the binary sequence 01100011. Compare it to the waveform that would be produced by ordinary phase modulation as described in problem 28.
b. Explain how differential phase modulation can be demodulated.

31. A new broadcast service is to transmit digital music using the FM radio band. Stereo audio signals are to be transmitted using a digital modem over the FM band. The specifications for the system are the following: Each audio signal is sampled at a rate of 40 kilosamples/second and quantized using 16 bits; the FM band provides a transmission bandwidth of 200 kiloHertz.
a. What is the total bit rate produced by each stereo audio signal?
b. How many points are required in the signal constellation of the digital modem to accommodate the stereo audio signal?

32. A twisted-wire pair has an attenuation of 0.7 dB/kilometer at 1 kHz.
a. How long can a link be if an attenuation of 20 dB can be tolerated?
b. A twisted pair with loading coils has an attenuation of 0.2 dB/kilometer at 1 kHz. How long can the link be if an attenuation of 20 dB can be tolerated?

33. Use Figure 3.37 and Figure 3.40 to explain why the bandwidth of twisted-wire pairs and coaxial cable decreases with distance.

34. Calculate the bandwidth of the range of light covering the range from 1200 nm to 1400 nm. Repeat for 1400 nm to 1600 nm. Keep in mind that the speed of light in fiber is approximately 2×10^8 m/sec.

35. Compare the attenuation in a 100 km link for optical fibers operating at 850 nm, 1300 nm, and 1550 nm.

36. A satellite is stationed approximately 36,000 km above the equator. What is the attenuation due to distance for the microwave radio signal?

37. Suppose a transmission channel operates at 3 Mbps and has a bit error rate of 10^{-3}. Bit errors occur at random and independent of each other. Suppose that the following code is used. To transmit a 1, the codeword 111 is sent; to transmit a 0, the codeword 000 is sent. The receiver takes the three received bits and decides which bit was sent by taking the majority vote of the three bits. Find the probability that the receiver makes a decoding error.

38. An early code used in radio transmission involved codewords that consist of binary bits and contain the same number of 1s. Thus the two-out-of-five code only transmits blocks of five bits in which two bits are 1 and the others 0.

a. List the valid codewords.
b. Suppose that the code is used to transmit blocks of binary bits. How many bits can be transmitted per codeword?
c. What pattern does the receiver check to detect errors?
d. What is the minimum number of bit errors that cause a detection failure?

39. Find the probability of error-detection failure for the code in problem 38 for the following channels:

a. The random error vector channel.
b. The random bit error channel.

40. Suppose that two check bits are added to a group of $2n$ information bits. The first check bit is the parity check of the first n bits, and the second check bit is the parity check of the second n bits.

a. Characterize the error patterns that can be detected by this code.
b. Find the error-detection failure probability in terms of the error-detection probability of the single parity check code.
c. Does it help to add a third parity check bit that is the sum of all the information bits?

41. Let $g(x) = x^3 + x + 1$. Consider the information sequence 1001.

a. Find the codeword corresponding to the preceding information sequence.
b. Suppose that the codeword has a transmission error in the first bit. What does the receiver obtain when it does its error checking?

42. ATM uses an eight-bit CRC on the information contained in the header. The header has six fields:

First 4 bits: GFC field
Next 8 bits: VPI field
Next 16 bits: VCI field
Next 3 bits: Type field
Next 1 bit: CLP field
Next 8 bits: CRC

a. The CRC is calculated using the following generator polynomial: $x^8 + x^2 + x + 1$. Find the CRC bits if the GFC, VPI, Type, and CLP fields are all zero and the VCI field is 00000000 00001111. Assume the GFC bits correspond to the highest-order bits in the polynomial.
b. Can this code detect single errors? Explain why.
c. Draw the shift register division circuit for this generator polynomial.

43. Suppose a header consists of four 16-bit words: (11111111 11111111, 11111111 00000000, 11110000 11110000, 11000000 11000000). Find the Internet checksum for this code.

44. Let $g_1(x) = x + 1$ and let $g_2(x) = x^3 + x^2 + 1$. Consider the information bits (1,1,0,1,1,0).
a. Find the codeword corresponding to these information bits if $g_1(x)$ is used as the generating polynomial.
b. Find the codeword corresponding to these information bits if $g_2(x)$ is used as the generating polynomial.
c. Can $g_2(x)$ detect single errors? double errors? triple errors? If not, give an example of an error pattern that cannot be detected.
d. Find the codeword corresponding to these information bits if $g(x) = g_1(x)g_2(x)$ is used as the generating polynomial. Comment on the error-detecting capabilities of $g(x)$.

45. Take any binary polynomial of degree 7 that has an even number of nonzero coefficients. Show by longhand division that the polynomial is divisible by $x + 1$.

46. A repetition code is an $(n, 1)$ code in which the $n - 1$ parity bits are repetitions of the information bit. Is the repetition code a linear code? What is the minimum distance of the code?

47. A transmitter takes K groups of k information bits and appends a single parity bit to each group. The transmitter then appends a block parity check word in which the jth bit in the check word is the modulo 2 sum of the jth components in the K codewords.
a. Explain why this code is a $((K + 1)(k + 1), Kk)$ linear code.
b. Write the codeword as a $(k + 1)$ row by $(K + 1)$ column array in which the first K columns are the codewords and the last column is the block parity check. Use this array to show how the code can detect all single, double, and triple errors. Give an example of a quadruple error that cannot be detected.
c. Find the minimum distance of the code. Can it correct all single errors? If so, show how the decoding can be done.
d. Find the probability of error-detection failure for the random bit error channel.

48. Consider the $m = 4$ Hamming code.
a. What is n, and what is k for this code?
b. Find the parity check matrix for this code.
c. Give the set of linear equations for computing the check bits in terms of the information bits.

d. Write a program to find the set of all codewords. Do you notice anything peculiar about the weights of the codewords?

49. Show that an easy way to find the minimum distance is to find the minimum number of columns of **H** whose sum gives the zero vector.

50. Suppose we take the (7,4) Hamming code and obtain an (8,4) code by adding an overall parity check bit.

a. Find the **H** matrix for this code.
b. What is the minimum distance?
c. Does the extra check bit increase the error-correction capability? the error-detection capability?

51. A (7,3) linear code has check bits given by

$$\begin{aligned} b_4 &= b_1 + b_2 \\ b_3 &= b_1 \qquad + b_3 \\ b_6 &= \qquad b_2 + b_3 \\ b_7 &= b_1 + b_2 + b_3 \end{aligned}$$

a. Find the **H** matrix.
b. Find the minimum distance.
c. Find the set of all codewords. Do you notice anything peculiar about the set of codewords.

52. An error-detecting code takes k information bits and generates a codeword with $2k+1$ encoded bits as follows:

The first k bits consist of the information bits.
The next k bits repeat the information bits.
The next bit is the XOR of the first k bits.

a. Find the check matrix for this code.
b. What is the minimum distance of this code?
c. Suppose the code is used on a channel that introduces independent random bit errors with probability 10^{-3}. Estimate the probability that the code fails to detect an erroneous transmission.

53. A (6,3) linear code has check bits given by

$$\begin{aligned} b_4 &= b_1 + b_2 \\ b_5 &= b_1 \qquad + b_3 \\ b_6 &= \qquad b_2 + b_3 \end{aligned}$$

a. Find the check matrix for this code.
b. What is the minimum distance of this code?
c. Find the set of all codewords.

54. (Appendix 3A). Consider an asynchronous transmission system that transfers a sequence of N bits between a start bit and a stop bit. What is the maximum value of N if the receiver clock frequency is within 1 percent of the transmitter clock frequency?