

# High-Performance

# Communication

# Networks

SECOND  
EDITION

Jean Walrand  
Pravin Varaiya

University of California,  
Berkeley



Morgan Kaufmann Publishers  
San Francisco, California

*Sponsoring Editor* Jennifer Mann  
*Director of Production and Manufacturing* Yonie Overton  
*Production Editor* Heather Collins  
*Editorial Assistant* Karyn Johnson  
*Cover Design* Ross Carron Design  
*Text Design, Composition, and Illustrations* Windfall Software  
*Copyeditor* Carol Leyba  
*Proofreader* Erin Milnes  
*Indexer* Ted Laux  
*Printer* Courier Corporation  
*Cover credit:* © Lockyer, Romilly/The Image Bank

Morgan Kaufmann Publishers

*Editorial and Sales Office*

340 Pine Street, Sixth Floor  
San Francisco, CA 94104-3205

USA

*Telephone* 415 / 392-2665

*Facsimile* 415 / 982-2665

*E-mail* [mfp@mfp.com](mailto:mkp@mfp.com)

*Web site* <http://www.mfp.com>

© 2000 by Morgan Kaufmann Publishers

All rights reserved

Printed in the United States of America

03 02 01 00 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—with the prior written permission of the publisher.

**Library of Congress Cataloging-in-Publication Data**

Walrand, Jean.

High-performance communication networks / Jean Walrand, Pravin Varaiya.

—2nd ed.

p. cm.—(The Morgan Kaufmann series in networking)

Includes bibliographical references.

ISBN 1-55860-574-6

1. Computer networks. 2. Multimedia systems. 3. High performance computing.  
4. Asynchronous transfer mode. 5. Wireless communications systems. I. Varaiya, P.P.  
(Pravin Pratap) II. Title. III. Series.

TK5105.5 .W353 2000

621.382'1—dc21

99-047341

CIP

**The Morgan Kaufmann Series in Networking**

Series Editor, David Clark

*High-Performance Communication Networks, 2e*

Jean Walrand and Pravin Varaiya

*Computer Networks: A Systems Approach, 2e*

Larry Peterson and Bruce Davie

*Internetworking Multimedia*

Jon Crowcroft, Mark Handley, Ian Wakeman

*Understanding Networked Applications: A First Course*

David G. Messerschmitt

*Integrated Management of Networked Systems: Concepts, Architectures, and their Operational Application*

Heinz-Gerd Hegering, Sebastian Abeck, and Bernhard Neumair

*Virtual Private Networks: Making the Right Connection*

Dennis Fowler

*Networked Applications: A Guide to the New Computing Infrastructure*

David G. Messerschmitt

*Modern Cable Television Technology: Video, Voice, and Data Communications*

Walter Ciciora, James Farmer, and David Large

*Switching in IP Networks: IP Switching, Tag Switching, and Related Technologies*

Bruce S. Davie, Paul Doolan, and Yakov Rekhter

*Wide Area Network Design: Concepts and Tools for Optimization*

Robert S. Cahn

*Optical Networks: A Practical Perspective*

Rajiv Ramaswami and Kumar Sivarajan

*Practical Computer Network Analysis and Design*

James D. McCabe

*Frame Relay Applications: Business and Technology Case Studies*

James P. Cavanagh

For a list of forthcoming titles, please visit our website at

<http://www.mkp.com/publish/mann/networking.htm>

We dedicate this book  
to *Annie* and *Isabelle* and *Julie*,  
and to *Ruth* 

---

# Preface

Much has changed in the networking world since 1995 when we wrote the first edition. “Online” and “Web” joined “Internet” in the popular vocabulary. Cellular phones became as common as the telephone. The “fast” 56 Kbps modem introduced with much publicity in 1998 was quickly surpassed by the megabit-per-second access delivered at home by cable TV and ADSL. And at work, 100 Mbps Ethernet came to the desktop.

The ongoing process of network convergence today is seen in the multi-billion-dollar acquisitions of cable TV operators and data networks by telephone companies. Service providers and equipment manufacturers are beginning to compete in the delivery of quality of service or QoS. That competition will shape the future of ATM and IP.

Advances in wireless communication promise soon to bring “anytime, anywhere” connectivity. Within a decade optical networking will provide orders of magnitude increases in bandwidth. These advances will sustain the networking boom of the 1990s. The social consequences of these developments are difficult to predict, but the technological trends are in place. We wrote the second edition to explain these and related technological advances, some unexpected, others already evident in 1995.

## Audience

This book is a uniquely comprehensive study of the major communication networks: data, telephone, cable TV, and wireless. We describe the technologies that help create these networks, explain the protocols and control mechanisms that operate them, and analyze the economic principles that regulate their use

and evolution. We wrote the book for the professionals and students who want such a comprehensive view of networking.

The professionals are those in industry who must evaluate their decisions in the context of the wide role that networking plays in their organizations. They may be networking engineers and computer scientists and their managers, corporate network managers and administrators, operations research and system engineers engaged in network design and operations or in upgrading networking infrastructure. Their decisions often will require an understanding of alternative technologies and performance evaluation and a sense of the pace and direction of innovation. We believe this book will help gain such an understanding.

College seniors and graduate students come to networking with training in electric engineering, computer science, or operations research. They are attracted to the field because of its career opportunities or because a familiarity with networking is now necessary for their own specialization in communications or software engineering or control. They may have taken at most one or, more likely, no undergraduate course in networking. This book will meet the diverse needs of these students and give them a wider, more sophisticated appreciation of this exciting field than other books with a narrow view of networking.

The distinction between these two intended audiences is only nominal. Today's professional was yesterday's student, and the astonishing pace of technical change will tomorrow make him or her a student again. To cope well with that pace requires a comprehensive view, and that is what we believe this book offers.

## Approach

We have conducted research in networking for twenty years. For the past fifteen years we have taught an introductory graduate course in networking to students from electrical engineering, computer science, and operations research. For more than ten years we have offered short courses to professionals from the telecommunications industry and to managers in charge of networking in their companies. For the past five years we have had intense interactions with industry, as consultant and technical adviser (PV, JW) and as entrepreneur (JW). This experience in research, teaching, and industry has shaped our book. In all three contexts we find the need for comprehensiveness of coverage and multiple perspectives.

Most books take a narrow view of the subject and approach networking from a single perspective. Typically, it is identified with the Internet or ATM

networks and described through the associated protocols. Or networks are modeled as networks of queues, whose operation is explained through routing algorithms and queuing analyses. Or networks are described through their enabling technology: wireless communication, optics, or switching.

We present a comprehensive study, discussing networks as the need arises from the basis of first principles from communications engineering, computer science, operations research, and economics. We have minimized the use of advanced concepts from these disciplines. It is our hope that the reader can thus gain a greater appreciation of these multiple views and a deeper understanding of how networks are built, how they are used, and who will pay for them. We discuss questions of network performance and control in an intuitive manner and, in a separate chapter, we present the rigorous mathematical argument.

## Highlights of the Second Edition

In addition to changes and updates we have made throughout the manuscript, we would like to highlight four major enhancements in the new edition. First, in the previous edition the Internet was treated simply as an example of packet-switched networks. There is now a complete study of the Internet, including the TCP/IP protocol suite and the advances proposed to improve its performance or provide quality of service.

Second, wireless communication, absent from the previous edition, now receives an extended discussion. The growing importance of wireless telephone access and its potential for use in data transfer mandated its inclusion.

Third, rapid advances in the last five years in wave-division multiplexing and wave-selective switching have brought forward the era of optical networking. These advances will eventually change the fundamentals of network design, operations, and economics, and so they are described here.

Lastly, quality of service (QoS) is likely to become an important dimension of competition among providers. The ability to operate networks that can give QoS guarantees is also key to service integration. The economics of QoS and the mechanisms needed to guarantee QoS receive much attention in the new edition.

## Contents

We give a chapter by chapter outline, pointing out the changes in the new edition. Chapter 1 contains a brief historical account and explains the principles

of networking. Added are recent estimates of the size, growth, and trends in the telecommunications industry. Chapter 2 explains how network services are produced by layered architectures. A new section summarizes applications that are driving networking demand.

Chapter 3 discusses packet-switched networks using the OSI model, and the important LAN implementations. The 100-Mbps Ethernet, and the replacement of Ethernet hubs by intelligent Ethernet switches that can create virtual local area networks or VLANS, have reorganized enterprise networking. Descriptions of these innovations and gigabit Ethernets are added.

A unified treatment of the Internet and TCP/IP networks occupies Chapter 4. Advances in Internet technology in addressing, faster switching, improvements in the TCP/IP protocol suite, and protocol proposals that seek better control are discussed.

Circuit-switched networks is the subject of Chapter 5. SONET continues to receive emphasis. The most significant addition is the discussion of broadband access networks: cable TV and ADSL, and European proposals advancing passive optical networks. Widespread deployment of these technologies will spur commercial development of broadband services.

Chapter 6 updates the explanation of ATM with important recent work, including internetworking protocols MPOA, and more detailed specifications of PNNI routing and UNI signaling. Much of this work is focused on more efficient ATM support of IP. How ATM and IP will compete and cooperate to provide QoS remains unresolved.

Wireless access has exploded worldwide over the last five years. Primarily used for voice and short message transfers, wireless communication is beginning to be used for data. Chapter 7 explains the characteristics of wireless links and the challenges these characteristics pose for networking. The discussion explains why, unlike the convergence experienced in wireline networks, wireless networking is fragmented and widespread adoption of wireless technology for data remains uncertain.

Chapter 8 provides an accessible discussion, and Chapter 9 explains the mathematical derivations, of network performance and control. The treatment covers circuit-switched, packet-switched, and ATM networks. Resource allocation (bandwidth and priority assignment) to achieve QoS guarantees using window and rate control algorithms are discussed there. The treatment of congestion control is novel.

Chapter 9, devoted to economics, now has a focus based on a formulation of demand for network services. Implementation of QoS guarantees will require pricing of QoS-differentiated services—a major departure from the current practice of flat-rate tariffs for network access. There are analyses of data about

how users value service quality in terms of their willingness to pay. The data are obtained from a market trial at Berkeley that began in April 1998.

Five years ago, wave-division multiplexing (WDM) was limited to laboratory demonstrations. Today, backbone optical links are being upgraded by installing WDM equipment. WDM links with 1 terabit per second speed (equal to the traffic carried by the entire Internet today) will be sold next year. Advances in optical routing and switching in less than ten years will culminate in all-optical networks, offering orders of magnitude higher speeds with a small increase in cost. This could inaugurate another revolution in communications. WDM and optical switching are discussed in Chapter 10. The treatment of optical links in the first edition has been abridged.

Chapter 11 updates the discussion on fast packet switching to incorporate multicasting and some recent work on fast table search. Chapter 12 gives a revised version of the future of networking.

## How To Use This Book

This book can be used by industry professionals, or as a text for undergraduate or graduate students. Professionals may study a topic as they need it to facilitate understanding of a particular development. An interesting undergraduate course can be taught around Chapters 1 through 3 and either Chapters 4 and 6, if the audience is primarily from computer science, or Chapters 5 and 7, if the students are primarily from electrical engineering.

We ourselves have used this material in two ways. At Berkeley, we have taught a one-semester, 45-hour introductory graduate course to students from electrical engineering, computer science, and operations research. (The course always attracts some seniors.) Students need no prior exposure to communication networks—the emphasis is on descriptive breadth that conveys the excitement of the technological advances and the challenges posed by speed, distance, and demanding applications. Three or four times each year we have taught a short course to practitioners, between 8 and 20 hours long. The aim there is to provide an overview of recent developments, to decipher trends, and to speculate about opportunities.

## Support Materials

Our own lectures make heavy use of the figures in the book. Postscript files of the figures are available from the Web page for our book at <http://www.mkp.com>.

Each chapter of the book ends in problems that test understanding of the material and challenge the reader to use that understanding in situations that may arise in practice. We will keep adding to these problems and post them at the Web site. A solutions manual is also available from the publisher.

## Acknowledgments

This book synthesizes the different viewpoints of networking specialists who know more about each view than we do. Inevitably, errors of fact and judgment and balance of treatment have crept into the book. We would be very grateful to our readers for bringing those errors to our attention and for providing us with feedback about their experiences in learning or teaching from this book. We can be reached via e-mail at *{wlr, varaiya}@eecs.berkeley.edu*. We will post corrections and comments at the Web site <http://www.mkp.com>. In this second edition we have incorporated comments from instructors who have used the first edition.

Andrea Goldsmith's chapter on wireless communications discusses a very important technology that was entirely missing in the first edition. We are greatly indebted to her for the excellent discussion of a rapidly evolving field. She can be reached at *andrea@ee.stanford.edu*.

A draft of the entire manuscript for the second edition was reviewed by Vijay Bhagavath, AT&T Labs; Scott Jordan, Northwestern University; Ivy Hsu, Nortel Networks; and Ramesh Rao, UC, San Diego. Anthony Ephremides, University of Maryland, reviewed Chapter 7; Kevin Fall, UC, Berkeley, reviewed Chapter 4; Riad Hartani, Nortel Networks, reviewed Chapter 6; and Eytan Mediano, MIT Lincoln Laboratory, reviewed Chapter 10.

We are immensely grateful to them for their criticisms as well as their suggestions for improving the book. Most of those suggestions have been incorporated.

Our editor, Jennifer Mann, provided the encouragement and friendly coaxing that we needed to start work on the second edition and to bring it to completion. Her assistant, Karyn Johnson, helped with logistics and with her enthusiasm. Finally, our thanks to our production editor, Heather Collins, who somehow managed a very tight schedule.

# Overview

**I**nformation technology is changing the world economy, society, and daily life. The change proceeds in waves, originating with the new technologies, growth, and restructuring of the affected industries. Next come changes in other businesses as they absorb those technologies. New businesses are started, and traditional practices are overthrown. Societies and governments try to adapt to shifts in the demand for goods and services, investment, and skilled workers. Millions of us change our daily routines in work and recreation and the way we interact with others, as we adapt, willingly or under pressure, to the new opportunities of the World Wide Web.

**Telecommunications Industry** According to the International Telecommunication Union (ITU), the total trade in telecommunications equipment and services worldwide reached \$1 trillion in 1998. The trade is growing at 7% annually, twice the rate of the world economy. International traffic doubled between 1990 and 1996 to 70 billion telephone minutes.

Technological advances in fiber optic communications and increased competition are dramatically reducing costs of raw bandwidth. According to British Telecom, equipment and installation cost per voice path on the transpacific route fell from \$73,000 in 1975 to \$2,000 in 1996, \$200 in 1999, and is likely to reach a mere \$5 in 2010. The average charge for a three-minute peak-rate U.S.-Europe phone call fell in five years from \$4 to \$1.50 in 1998 and will reach 50¢ by 2000. Demand for access is insatiable. In 1996, forty-eight million new fixed lines were installed bringing the global installed base to 741 million. The

number of cellular subscribers grew by 50% in 1997 to reach 200 million in 1998.

The traffic mix is changing. The number of computers on the Internet grew from 1 million in early 1993, to 5 million in 1995, 16 million in 1997, and over 50 million in 1999. The Internet Society estimates that the number of Internet users will reach 300 million by the end of 2000. That is 5% of the world population. By 2047 the world population will reach 11 billion, and if 25% becomes connected to the Internet, that is nearly 3 billion people. Data traffic in the transatlantic corridor is doubling each year and surpassed voice traffic in September 1997. By 2000, it may account for 75% of all traffic. The character of the 100-year-old telephone industry will change as it attempts to meet the shift in demand.

Under pressures from users, suppliers, and a worldwide movement for deregulation, telecommunications monopolies are ending. The 1997 World Trade Organization agreement on telecommunications services, signed by 69 countries with more than 90% of all telecommunications trade, calls for the liberalization of nearly all markets by 2000 or soon after. In Europe, the opening of most markets in January 1998 brought in its wake new entrants and large investments. This was foreshadowed by the 1996 Telecommunications Act in the U.S.

The impact of the 1996 Act is still working itself out. AT&T, the world's largest long-distance carrier, set a new course for itself, when it acquired two large cable TV operators, TCI and MediaOne, as part of its strategy to be an integrated provider of TV, data, and phone services. A group of on-line service providers is challenging AT&T, urging Congress and the FCC to extend the Act to include "open access" to cable TV networks. The reported \$58 billion price tag for MediaOne, amounting to \$10,000 for each of its five million customers, suggests that AT&T is anticipating an annual revenue of at least \$2,000 per customer. The regional telephone companies are seeing their monopolies erode by competitive local exchange carriers or CLECs and by cable TV companies. Several of these companies are merging (NYNEX/Bell Atlantic, SBC/Pacific Bell/Ameritech) as part of a strategy to enter long-distance markets in exchange for their monopoly position in local markets.

A less noticed but potentially significant change may be initiated by the use of ATM switches in place of the large and expensive time-division circuit switches used in telephone networks everywhere. ATM switches are cheaper, can be deployed in smaller sizes, and are more versatile since ATM supports any form of traffic. Telephone companies, burdened by their large installed infrastructure and monopoly heritage, are likely to be further threatened by new entrants that offer ATM-based services initially catering to niche markets.

Yesterday's niche telecommunications market may grow into a large market tomorrow. Mobile subscribers increased from 10 million in 1990 to 200 million in 1998. WWW applications have transformed business networks into intranets—internal corporate networks based on Internet protocols. Unknown in 1994, by 1997 over half of all large corporations had implemented intranets.

***Economy and Society*** Advances in telecommunications are felt in the rest of the economy. Long skeptical about the productivity benefits of information technology, many economists now attribute to it a productivity increase of 1%. For the \$8.5 trillion U.S. economy, this amounts to an additional \$85 billion of "free" output each year. Investment in computing and communications equipment has quadrupled over the last decade, accounting for 53% of all business spending on equipment. There are similar increases in spending on software, consulting, technical support, and training.

Some economists believe even this productivity growth is understated because government statistics do not adequately define and measure output in the fast-growing service sector, including banking, finance, health care, and education.

While "hot" Internet e-commerce companies gain media attention, it is in the mundane, back office operations of invoicing, purchasing, and inventory control that the impact of intranets and internets is most profound. Business-to-business commerce over the Internet is projected to jump from \$48 billion in 1998 to \$1.5 trillion by 2003, according to Forrester Research, Inc. During the same period, consumer sales over the Internet will rise from \$3.9 billion to \$108 billion.

Some believe that tomorrow's corporations will be "virtual"—defined not by their location but by their ability to acquire knowledge, organize information, and orchestrate independent contractors and suppliers worldwide. In the process, businesses and professions are disappearing. Customers are leaving travel agents and other retailing intermediaries, preferring to make their reservations and purchases on-line.

The world of work is changing. More than three-quarters of workers in the U.S. today are "information workers." There is a global shortage of workers in information management and technology. Spurred by high labor costs, U.S. companies are hiring programmers and engineers in India, Eastern Europe, and Russia. We may be witnessing the next phase in which capital searches for opportunities in India, Israel, and elsewhere. At the same time there is increased employment insecurity as corporations shift their demand for skilled labor in the face of competitive pressure to change their operations.

With more than a million pages added each day, the Web is now an infinitely large bulletin board. Media Metrix estimates that 61 million people worldwide visited the top 50 Web sites in the U.S. in April 1999. The Web is changing our habits. America Online's 14 million users are sending 15 million e-mails each day and spending on average 21 hours on-line each month. By comparison, the average person in the U.S. watches TV 25 hours per week.

***Problems and Opportunities*** Technologies open up possibilities that can create problems. The problems suggest opportunities for further advances. There are now several million Web sites. To navigate this worldwide bulletin board, search engines were invented. They crawl through these sites maintaining, for every word, a list of all Web pages containing that word. But this is not always satisfactory. A search for "modem" in Alta Vista returned 2 million pages, a search for "DSL modem" returned 800 pages, still too large. This creates the opportunity for search facilities that can better figure out what the viewer is looking for.

Some people are troubled by the growth of on-line pornography and violence. Technological solutions are being offered that provide services that rate Web sites and means to prevent access to undesired sites.

Telecommunications enables giant media companies to manufacture and spread a worldwide mass culture, erasing local boundaries and sensibilities. But the technology also gives opportunities for creative resistance. The copying machine and the fax machine brought ordinary citizens the means to publish and propagate their own views. Over the last 10 years telecommunications has vastly increased the reach and lowered the cost of disseminating news and ideas. Geographically isolated, like-minded individuals are forming small groups to pursue their common interests and to build new cultural islands outside mass culture. A movement such as this would have been impossible a decade ago. Further in the future lies the possibility that education will become a truly lifelong process, enabling a much fuller development of our potential.

It is difficult to predict the long-term impact of the information revolution. But the technologies that underlie the revolution can be understood using the language and concepts of the network engineer, the computer scientist, and the economist. These technologies comprise advances in computers, communications, signal processing, and their applications in diverse domains. The way these combine to form advances in networking form the substance of this book.

This book provides a system-level understanding of the networking technologies and the actual networks that promote the information revolution. You will learn that while there are many details, only a few principles are sufficient to grasp the field of networking. You will then know what questions to ask in

order to compare different networks, and in many instances you will know how those questions should be answered. If you work in an organization, you will have the basis to judge how well different networking solutions will meet the needs of the organization, now and in the future. If you are a student, you will be able to critically read many of the recent research contributions to networking, and you will gain the sense of what directions of research are likely to be fruitful.

Advances in networking technology feed on, and are constrained by, the current state of networking. Within those constraints, the advances are guided by wider technological and economic forces. This chapter presents a highly abbreviated history of the key innovations in telephone, data, cable TV, and wireless networks, and the principles that can serve as a compass for judging which directions of technological advance are likely to be more successful. By the end of the chapter you will be acquainted with the main contenders for the role of “network of the future,” and you will appreciate that no single network technology will be the winner: the future network will integrate all of the major networking solutions.

In section 1.1 we review the history of the telephone network, computer or data networks, cable television or CATV, and wireless networks. In the past, these networks used different technologies that were well suited to the information services they provided.

In section 1.2 we explain the four principles that underlie the forces driving the industry toward convergence and leading to the interpenetration of these networks.

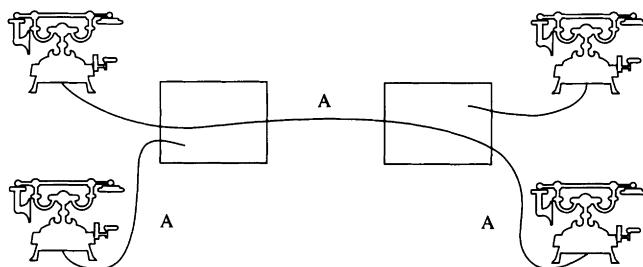
In section 1.3 we discuss some plausible scenarios for networks of the future. These future networks will offer services ranging from telephone to interactive video to high-speed file transfers.

---

## 1.1

### HISTORY OF COMMUNICATION NETWORKS

Communication networks enable users to transfer information in the form of voice, video, electronic mail or e-mail, and computer files. Users request the communication service they need by means of simple procedures using a telephone handset or cellular phone, set-top TV box, or through applications running on a host computer such as a PC or workstation. In the following sections, we identify the major steps in the evolution of communication



**1.1**  
**FIGURE**

Telephone network around 1890. The transmissions are analog, and the switches are manually operated.

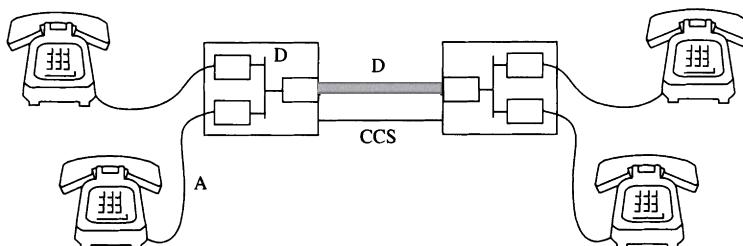
networks that provide the services users want. There were many other innovations, but the steps emphasized here were decisive.

### 1.1.1 Telephone Networks

The key innovations in telephony are circuit switching, digitization, separation of call control from voice transfer, optical links, and service integration.

In 1876, Alexander Graham Bell invented a pair of telephones. Around 1890, simple networks connected telephones by manually operated switches. In such a network, as shown in Figure 1.1, the signal is analog, as indicated by the letter A on the links. To call another telephone, a customer first rings the operator and provides the phone number of the other party. The operator then determines the line that goes either directly to the other party or to another operator along a path to the other party. In the latter case, the operators talk to each other, decide how to handle the call, and the procedure of constructing the path to the other party continues, possibly involving other operators. Eventually, one operator rings the destination, and, if the telephone is picked up, the two parties are connected. The parties remain connected for the duration of the conversation and are disconnected by the operators at the end of the conversation.

Note how the transmission lines are allocated to the phone conversation. This is accomplished by *circuit switching*, where “circuit” refers to the capability of transmitting one telephone conversation along one link. To set up a call, a set of circuits has to be connected, joining the two telephone sets. By modifying the connections, the operators can switch the circuits. Circuit switching occurs at the beginning of a new telephone call. Operators were later replaced by mechanical switches and, 100 years later, by electronic switches.



**FIGURE**  
1.2

Telephone network around 1988. The transmissions are analog (A) or digital (D). The switches are electronic and exchange control information by using a data network called common channel signaling (CCS).

Figure 1.2 illustrates the telephone network around 1988. One major development at this stage is that the transmission of the voice signals between switches is digital, as indicated by the letter D, instead of analog.

An electronic interface in the switch converts the analog signal traveling on the link from the telephone set to the switch into a digital signal, called a *bit stream*. The same interface converts the digital signal that travels between the switches into an analog signal before sending it from the switch to the telephone.

The switches themselves are computers, which makes them very flexible. This flexibility allows the telephone company to modify connections by sending specific instructions to the computer. Figure 1.2 also shows another major development—*common channel signaling* (CCS). CCS is a data communication network that the switches use to exchange control information. This “conversation” between switches serves the same function as the conversation that took place between operators in the manual network. Thus CCS separates the functions of call control from the transfer of voice. Combined with the flexible computerized switches, this separation of function facilitates new services such as call waiting, call forwarding, and call back.

In current telephone networks, the bit streams in the trunks (lines connecting switches) and access links (lines connecting subscriber telephones to the switch) are organized in the digital signal (DS) hierarchy. The links themselves—the “hardware”—are called *digital carrier systems*. Trunk capacity is divided into a hierarchy of logical channels. In North America these channels, listed in Table 1.1, are called DS-1, . . . , DS-4 and have rates ranging from 1.544 to 274.176 Mbps (megabits per second). The basic unit is set by the DS-0 channel, which carries 64 Kbps (kilobits per second) and accommodates one voice circuit. Larger-capacity channels multiplex several voice channels. The

Medium	Signal	No. of voice circuits	Rate in Mbps		
			North America	Japan	Europe
T-1 paired cable	DS-1	24	1.5	1.5	2.0
T-1C paired cable	DS-1C	48	3.1		
T-2 paired cable	DS-2	96	6.3	6.3	8.4
T-3 coax, radio, fiber	DS-3	672	45.0	34.0	32.0
Coax, waveguide, radio, fiber	DS-4	4032	274.0		

**1.1**  
**TABLE**

Digital carrier systems. This is the hierarchy of digital signals that the telephone network uses. Note that the bit rate of a DS-1 signal is greater than 24 times the rate of a voice signal (64 Kbps) because of the additional framing bits required.

rates in Japan and Europe are different. The most common channels are DS-1 and DS-3.

Observe in the table that the rates are not multiples of one another: the DS-1 signal carries 24 DS-0 channels, but its rate is more than 24 times 64 Kbps. The additional bits are used to accommodate DS-0 channels with rates that deviate from the nominal 64 Kbps because the signals are generated using clocks that are not perfectly synchronized.

Since the 1980s the transmission links of the telephone network have been changing to the SONET, or Synchronous Optical Network, standard. SONET rates are arranged in the STS (Synchronous Transfer Signal) hierarchy shown in Table 1.2. In North America and Japan the basic SONET signal, STS-1, has a rate of 51.840 Mbps. (In Europe the basic signal is STS-3 and has a rate of 155.52 Mbps. The hierarchy is called Synchronous Digital Hierarchy, or SDH.) The most common links in the backbone today are OC-3, OC-12, and OC-48. OC-192 links are now coming into service. Wave-division multiplexing now enables a single optical fiber to carry 100 OC-192 links for an aggregate rate of 1 terabit per second.

Two differences are immediately apparent when comparing the STS and DS hierarchies. First, SONET signals have much higher bit rates, thanks to the much higher rates that optical links can support compared with the copper links of the current network. Second, the STS-*n* rate is exactly *n* times the STS-1 rate. Because all clocks in a SONET network are synchronized to the same master clock, it is possible to compose an STS-*n* signal by multiplexing

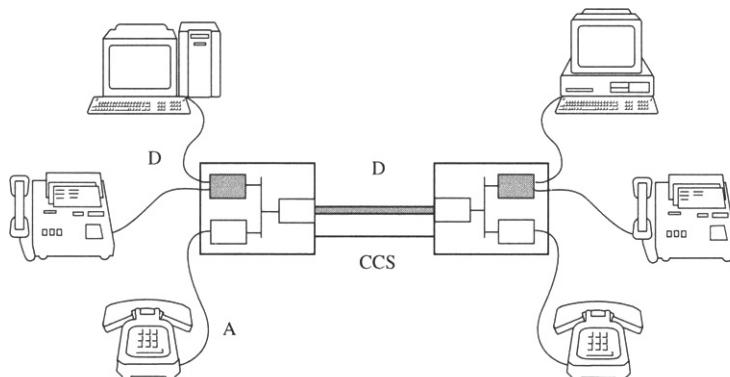
Carrier	Signal	Rate in Mbps
OC-1	STS-1	51.840
OC-3	STS-3	155.520
OC-9	STS-9	466.560
OC-12	STS-12	622.080
OC-18	STS-18	933.120
OC-24	STS-24	1244.160
OC-36	STS-36	1866.240
OC-48	STS-48	2488.320
OC-192	STS-192	9853.280

1.2  
TABLE

SONET rates. The rates of multiplexed STS-1 signals are exact multiples; no additional framing bits are used.

exactly  $n$  STS-1 signals. As a result, multiplexing and demultiplexing equipment for STS signals is less complex than for DS signals.

The last major innovation in telephony is the integration of voice and data signals through the introduction of the *Integrated Services Digital Network* (ISDN), illustrated in Figure 1.3. The ISDN basic access offered to a customer consists of two B channels and one D channel (both B and D channels are digital). Each B channel is a bidirectional, or full-duplex, channel at 64 Kbps. One B channel can carry either a circuit-switched connection, a packet-switched



1.3  
FIGURE

Integrated Services Digital Network (ISDN). The basic access provides two bidirectional 64-Kbps links and one 16-Kbps link. These links can be used to transmit voice or data.

transmission service (described below), or a permanent digital connection. The D channel carries a 16-Kbps packet-switched service. ISDN makes available to subscribers the digital transmission facilities that were previously used between the switches of the network, thus extending the digital transmission all the way to the users. Applications of the ISDN services include computer communication, high-speed facsimiles, remote monitoring of buildings, videotex, and low bit rate videophones. With ISDN, the telephone system is transformed into a network that can transfer information in many forms, if at modest speeds.

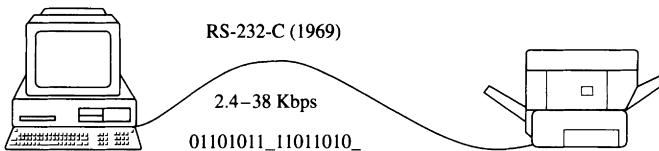
A new technology for transmission of data over untwisted or twisted pair cables for distances up to 4,000 m will displace ISDN. The technology uses existing telephone subscriber lines in a manner similar to ISDN. Although the telephone voice channel is limited to a bandwidth of 3 kHz, the twisted pair cable itself which connects to the central office has a bandwidth of more than 1 MHz, limited by signal attenuation and noise. Asymmetric Digital Subscriber Line (ADSL) service now offered by the telephone companies can provide up to 1.5 Mbps or more downstream (to the home) and up to 1.5 Mbps upstream (from the home), in addition to regular analog telephone service. It is estimated that 66% of subscriber loops in the United States can support the ADSL technology.

### 1.1.2 Computer Networks

This section discusses the following key innovations in computer or data networks: organization of data in packets, packet switching, the Internet Protocol hierarchy, multiple access methods, and service integration.

We begin our historical sketch in 1969 with the RS-232-C standard for the *serial port* of computer devices, illustrated in Figure 1.4. This standard is for low bit rate transmissions (up to 38 Kbps) over short distances (less than 30 m). Serial transmission proceeds one character at a time. The computer devices encode each character into seven bits, to which they can add a parity bit for error detection, and successive characters are separated by some time interval. When the receiver detects the beginning of a new character, it starts a clock that times the subsequent bits. Both bit rate and distance must be kept small, because transmissions take place over untwisted wires, which can introduce errors due to cross-talk. Cross-talk becomes more severe as the rate and the distance increase.

A serial link is often used to attach a computer to a *modem*. A modem, or modulator-demodulator, transmits data by converting bits into tones that can be transported by the telephone network as if they were voice signals. The receiving modem then converts these tones back into bits, thus enabling two computers with compatible modems to communicate over the telephone



1.4

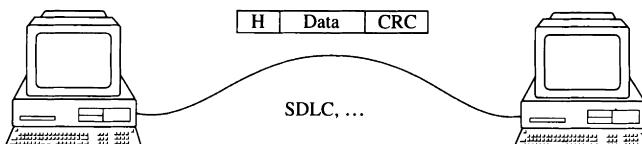
**FIGURE**

The RS-232-C standard for the serial line specifies the transfer of one 8-bit character at a time, separated by time intervals. The speed and distance of the serial line are limited.

network as if they were directly connected by a serial link. In 1999 most modems run at a speed of 28,800 bps. Modems conforming to the new V.90 standard can transmit 56,000 bps in the downstream direction.

Figure 1.5 illustrates the *synchronous transmission* standards introduced in the 1970s to increase the transmission rate and the usable length of transmission links. These standards are known as SDLC (Synchronous Data Link Control). A number of standards are based on SDLC, including HDLC (High-Level Data Link Control), LAPB (Link Access Procedure B), LAPD, and LAPS. The main idea of SDLC is to avoid the time wasted by RS-232-C caused by gaps between successive characters. To eliminate that lost time, SDLC groups many data bits into *packets*. A packet is a sequence of bits preceded by a special bit pattern called the *header* and followed by another special bit pattern called the *trailer*. The number of bits in a packet may be fixed or variable.

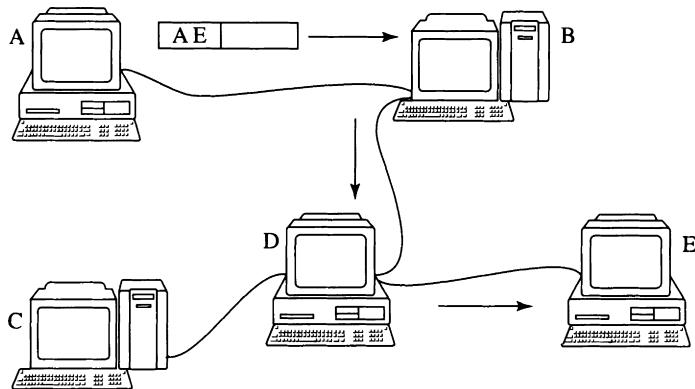
The receiver is synchronized by a preamble contained in the header (H) of the packet and by a self-synchronizing code that contains the timing information in addition to the data. Moreover, SDLC uses an error-detection code called the *cyclic redundancy check*, or CRC, that is more efficient and more powerful



1.5

**FIGURE**

The Synchronous Data Link Control and related standards transmit long packets of bits. The header (H) contains the preamble that starts the receiver clock, which is kept in phase by the self-synchronizing encoding of the bits. The receiver uses the cyclic redundancy check (CRC) bits to verify that the packet is correctly received.



**FIGURE**  
1.6

Store-and-forward transmissions proceed by sending the packet successively along links from the source to the destination. The packet header specifies the source and destination addresses (A and E, for example) of the packet. When it receives a packet, a computer checks a routing table to find out on which link it should next send the packet.

than the single parity bit of RS-232-C. Two computers, then, can exchange information over a transmission link using either RS-232-C or SDLC. But what if many computers are to be interconnected? In the early 1960s, communication engineers proposed the *store-and-forward packet-switching* method illustrated in Figure 1.6.

This figure shows computers connected by point-to-point links. To send a packet to computer E, computer A puts the source address A and the destination address E into the packet header and sends the packet to computer B. When B gets the packet from A, it reads the destination address and determines that it must forward the packet to D. When D gets the packet, it reads the destination address and forwards the packet to E. In this scheme, when a node receives a packet, it must first store it, then forward it to another node (if necessary). Hence the name *store-and-forward* given to this switching method.

When computers use store-and-forward packet switching, they use a given link only when they send a packet. As a result, the same links can be used efficiently by a large number of intermittent transmissions. This method for sharing a link among transmissions is called *statistical multiplexing*. Statistical multiplexing contrasts with time-division multiplexing-based circuit switching, which reserves circuits for the duration of the conversation even though the parties connected by the circuit may not transmit continuously.

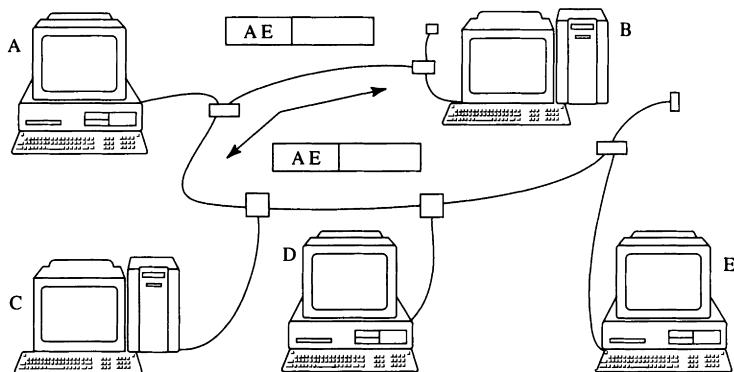
Starting in the late 1960s, the U.S. Department of Defense Advanced Research Projects Agency (DARPA) began promoting the development of packet-switched networks. The resulting network, ARPANET, began operations in 1969 by connecting four computers. The rules of operations, or protocols, ARPANET used were published in the open literature. By implementing these protocols, engineers in many research and educational institutions attached their computers to the ARPANET.

Through the ARPANET protocols, engineers agreed on a single packet format standard and a common addressing scheme. This allowed networks that conformed to this packet format to be easily interconnected. The benefits of interconnectivity soon became obvious, and the ARPANET evolved into the Internet, which today is used to interconnect a large number of computers and local area networks throughout the world. In 1983, only 500 "host computers" had Internet access. In 1999, there were 50 million such computers being used by 300 million people.

The single packet format of the Internet Protocol offered two advantages. On the one hand, the format could be supported by a variety of physical networks, including local area networks (LANs) such as Ethernet and token ring, as well as by point-to-point links. On the other hand, engineers and computer scientists could develop communications applications assuming that data would be transported in packets of a standardized format. The ARPANET implicitly implemented a three-layered architecture consisting of (1) the physical network that transfers bits, (2) groups of data encapsulated into packets with a common format and addressing scheme, and (3) applications that assume transfer of packets with no regard to the underlying physical network. This implicit layered architecture was subsequently elaborated and formalized in the Open Systems Interconnection, or OSI, model.

In the late 1960s and early 1970s, engineers proposed a new method for connecting computers. This method is called *multiple access*. It dramatically reduced the cost of interconnecting nearby computers in a LAN as well as the cost of access to a wide area network (WAN).

Figure 1.7 illustrates a popular implementation of multiple access called *Ethernet*. In an Ethernet network, computers are attached to a common coaxial cable via an interface that today consists of a small chip set mounted on the main board. When computer A wants to send a packet to computer E, it puts the source address A and the destination address E into the packet header and transmits the packet on the cable. All the computers read the packet, but only the computer with the destination address indicated on the packet copies it. The original Ethernet transmission rate was 10 Mbps; now 100-Mbps and 1000-Mbps Ethernet are available.



1.7

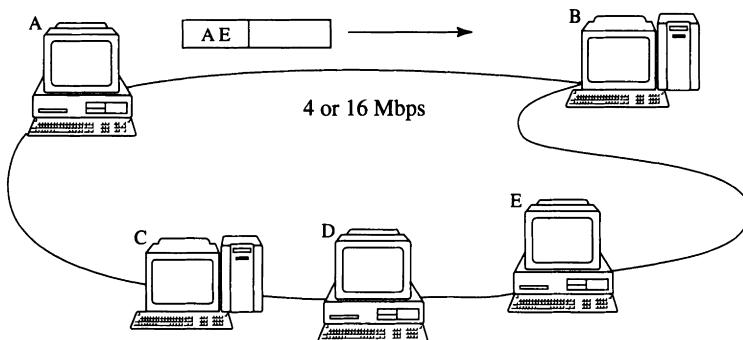
**FIGURE**

Ethernet. In this network, computers are attached to a common coaxial cable. The computers read every transmitted packet and discard those not addressed to them.

In the early 1980s, IBM developed another multiple access method, called *token ring*, illustrated in Figure 1.8. When networked as a token ring, computers are attached by point-to-point links in a unidirectional ring configuration using token ring interface boards. When the computers have no information to transmit, the interfaces pass a *token* around the ring. The interface boards between the computers and the network are configured so that they put back on the ring whatever information they receive with a delay equal to a few bit transmission times. This enables the token to circulate very fast around the ring.

Suppose computer A wants to send a packet to computer E. Computer A puts the source address A and the destination address E into the packet header and gives the packet to its interface, which then waits for the token. As soon as it gets the token, the interface of computer A transmits its packet, instead of forwarding the token. The other computers keep forwarding the packet they receive while making a copy for themselves. In particular, the interface of computer E copies the packet destined to it. The other interfaces discard their copy of the packet when they find out that it is not for them. When A receives the last bit of its own packet, after the packet has traveled around the ring, it puts the token back on the ring. What is important is that the computers get to transmit in turn, when they get the token. Hardware is available for token ring networks at 4 Mbps and at 16 Mbps.

The maximum time a computer waits before it gets to transmit in a token ring or Ethernet network is small enough for many applications but too large for

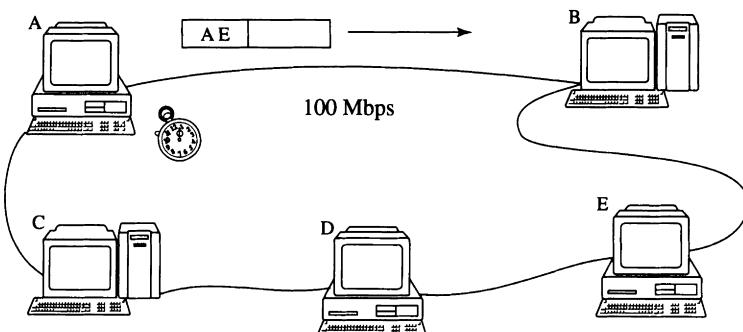


1.8

**FIGURE**

Token ring. The computers share a ring. Access is regulated by a token-passing protocol.

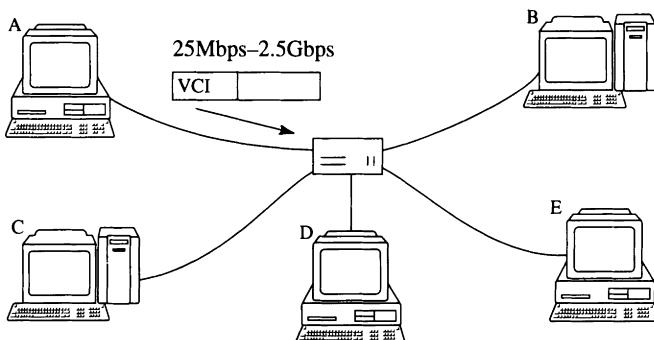
interactive audio or video applications. Also, the transmission rate of token ring (4 or 16 Mbps) or 10-Mbps Ethernet networks is too slow for some multimedia applications. These two limitations led engineers in the late 1980s to develop a new network called *Fiber Distributed Data Interface* (FDDI), illustrated in Figure 1.9. FDDI networks use optical fibers to transmit at 100 Mbps; access to the channel is regulated by a timed-token mechanism. This mechanism is similar to the access control of a token ring network except that with FDDI, the arrivals of tokens are timed to assure that they are retransmitted within a fixed time.



1.9

**FIGURE**

Fiber Distributed Data Interface (FDDI). A token-passing protocol is used to share the ring. The computers time their holding of the token. This network guarantees that every computer gets to transmit within an agreed-on time.



**1.10**  
**FIGURE**

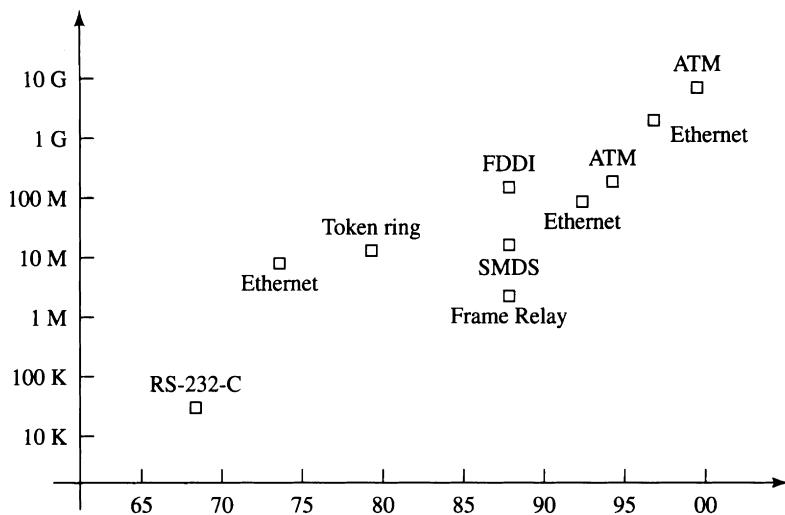
Asynchronous Transfer Mode (ATM) network. The network transports information in 53-byte cells. Total throughput of this network is much larger than that of FDDI or of a 100-Mbps Ethernet.

The high speed of FDDI makes it suitable for networking workstations with instruction rates of a few hundred Mips (millions of instructions per second). Because it can guarantee a token rotation time, FDDI can offer integrated services for applications that combine audio and video with data.

Faster LAN and WAN networks are today being deployed, using *Asynchronous Transfer Mode* (ATM). With ATM, a computer transmits information at rates between 25 Mbps and 2.5 Gbps (gigabits per second) in packets of 53 bytes (1 byte = 8 bits). These fixed-size packets, called *cells*, can be switched rapidly by ATM switches. Figure 1.10 illustrates an ATM LAN network with one switch. The header contains a virtual circuit address or VCI, instead of a source and destination address.

With the appropriate control software, network engineers can connect many ATM switches together to build large networks. Moreover, the links between ATM switches can be long optical fibers. Using this technology, then, companies can build a worldwide network. In an ATM network, data is transferred from source to destination over a fixed route, just like in a telephone connection. Unlike telephone networks, however, an ATM connection is not allocated a fixed bandwidth. The ATM network determines how much bandwidth to allocate so that information is transported with very low loss rate or delay, as required by the application. Thus this technology is well suited for building large integrated services networks.

Figure 1.11 summarizes the increase in speed of data networks. Over the 30 years since 1970, speed has increased by six orders of magnitude, from 10 Kbps to 10 Gbps.



1.11

FIGURE

The speed of data networks increased by six orders of magnitude between 1970 and 1999.

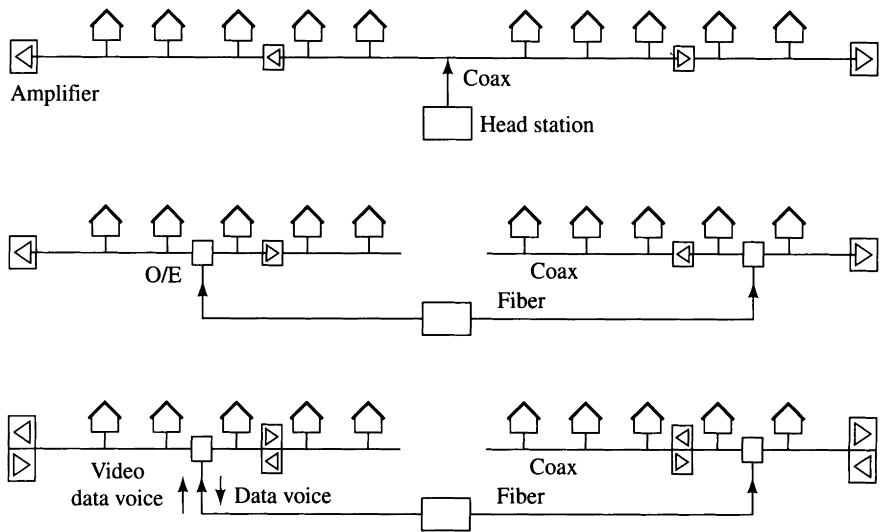
### 1.1.3 Cable Television Networks

Cable television, originally known as *Community Antenna Television* or *CATV*, was introduced in the 1940s in areas that could not receive the TV signal without obstruction. The solution was to place an antenna on top of a large utility pole and locally share the signal. CATV, as we now know it, was created when the signal from one master antenna was distributed over a large area using coaxial cable and amplifiers. The key innovations in cable TV are optical feeder links, digital compression techniques, and service integration.

Cable television systems deliver television signals to more than 50% of U.S. households and could serve close to 90% with relatively minor additional infrastructure. Changes in CATV networks are depicted in Figure 1.12.

Today, CATV uses frequency-division multiplexing to transmit up to 69 analog TV channels, each 4.5-MHz wide. Transmission is over coaxial cables arranged as a unidirectional tree, with wideband amplifiers used to compensate for the attenuation of the cable signal (top panel in Figure 1.12). The number of TV channels is limited by the bandwidth of coaxial cables. The span of a CATV network is limited by the noise power, which increases as more amplifiers are added to compensate for the signal power loss during propagation.

The next major step in CATV is to utilize optical fibers to transmit the TV signals over longer distances (middle panel in Figure 1.12). Fibers have



1.12

FIGURE

CATV networks have improved in two steps. In the first step the coax distribution system is replaced by fiber. In the second step channels are provided from the user to the head station.

a much lower attenuation than coaxial cables, so they can transmit signals over longer distances before it becomes necessary to use an amplifier. In this implementation, the transmission over the fiber is still analog. The signal is fed into the coaxial cable network at various points, where the optical signal is converted into electrical signals (indicated by the box labeled O/E). The cost of each optical transmission line is spread over a few hundred users. Moreover, existing coaxial cables can be reused. This *hybrid fiber/coaxial* (HFC) cable distribution system has a longer span and better signal quality than a coaxial cable network. The network is now a tree whose first level is a fiber network and whose bottom levels are coaxial cable. (In the top panel of Figure 1.12, both levels of the tree are coax.) This network is also called a *fiber-to-the-curb* (FTTC) network, where “curb” designates a location in some neighborhood where the fiber is connected to the local coaxial distribution network.

To increase the number of TV channels, the CATV industry is now migrating to a digital transmission technology. Before transmitting the TV signals, the CATV company uses a TV codec (coder-decoder) that converts each signal into a bit stream that represents the video frames. Using compression algorithms that have been standardized by the *Motion Pictures Expert Group* (MPEG), the codec compresses the bit stream to reduce its rate. This is accomplished by eliminating redundant information as well as information that does not con-

tribute significantly to the image quality, as perceived by the viewers. The bit streams are transmitted over fibers to the curb and are then distributed by the neighborhood coaxial network. The compression gain now allows the network to transmit about 500 TV channels. Using the first version of the MPEG standard, MPEG1, a moderate-quality TV signal is encoded as a 1.5-Mbps bit stream, which can be modulated in a signal that has a bandwidth of about 600 kHz. (By comparison, the analog NTSC TV signal has a bandwidth of 4.5 MHz, about seven times larger. NTSC is the North American standard for commercial broadcast color TV.) The decompression is performed by set-top boxes at the user residence. This CATV network is still unidirectional.

To provide new services, such as video on demand, Internet access, and telephony, the CATV industry is deploying bidirectional networks. Such a network (depicted in the bottom panel of Figure 1.12) connects video servers to users by means of control messages. The user chooses these messages to select the video program, and the video program is sent over the network to the user. A cable modem can give users access to a shared 3 Mbps (likely to increase to 10 Mbps) upstream channel to the Internet. Forrester Research, Inc., estimates that 2 million cable TV connections will be equipped with cable modems by 2000. The upstream bandwidth can also be used for Internet access and telephone service.

ADSL, noted in section 1.1.1 provides a cost-effective alternative to the use of the CATV network for data transmission.

### 1.1.4 Wireless Networks

Modern wireless communication dates back to Marconi's first radio transmission in 1895. Commercial radio stations in the U.S. were established by 1920. The first commercial TV programs were broadcast in 1941, color TV aired in the mid 1960s, and the first HDTV (high-definition TV) station began in 1998. These are all one-way, broadcast transmissions.

In 1946 public mobile telephone service was introduced in 25 U.S. cities. The system used a central transmitter that broadcast over the entire city. Because only one transmission at a time was permissible, system capacity was very limited. The solution to the capacity problem emerged in the 1960s, based on the concepts of cells and frequency reuse. The idea was to take advantage of the fact that a radio signal attenuates very rapidly: take the limited transmitter power, divide the city into "cells," and reuse the same frequencies by simultaneous transmissions in nonadjacent cells. The original system was analog. The current generation systems are all digital. To meet the huge demand, the industry is reducing the cell size, discovering more efficient modulation schemes,

while governments are turning over more spectrum for wireless communications. Mobile telephony is an extension of the wireline telephone system, and calls are circuit-switched. The growth of mobile telephony has been extraordinary, reaching 200 million subscribers worldwide by early 1998.

The first packet-switched wireless network was developed in 1971 at the University of Hawaii, under the sponsorship of DARPA. Alohanet, as the network was called, interconnected computers on four islands in a star topology: two computers could exchange packets through a central computer hub. Wireless local area networks permitting peer-to-peer communications networks are now available with bit rates on the order of 2 Mbps. However, unlike mobile phones, which have achieved a penetration comparable to wireline phones, wireless LANs have barely achieved a toehold in the market. The need for wireless LANs may grow with the demand for notebook and palm-sized computers.

Unlike wireline networks, whose backbone speeds will soon exceed 1 terabit per second, bandwidth in wireless networks remains a very scarce resource. The speed gap between wireline and wireless access will continue to grow with 100-Mbps Ethernets now common while 2-Mbps wireless LANs still are rare. The challenge for wireless networks will be to provide connectivity to the Internet, integrated services, especially voice and data, as well as multimedia Web access.

The key innovations in telephone, computer, CATV, and wireless networks are summarized in Table 1.3. The differences among these types of networks are still great. However, the table reveals that each type of network is now able

Telephone networks	Computer networks	Cable TV	Wireless networks
Circuit switching, CCS, separation of call control from voice transfer	Packet-switched networks, multiple-access networks	Digitization and compression using signal processing techniques	Radio and TV broadcast
ISDN and service integration	Layered architecture, ARPANET	Fiber-to-the-curb network	Cellular telephones
Optical links, SONET	Internet, OSI model	Two-way links	Wireless LANs
ATM	Integrated services, ATM	Service integration	Voice, data integration

1.3

TABLE

Key innovations in telephone, computer, and CATV networks.

to provide services that were formerly the exclusive province of other networks. We can discern in this the tendency toward “convergence.” In the next section we study the forces underlying this tendency.

## **1.2** NETWORKING PRINCIPLES

---

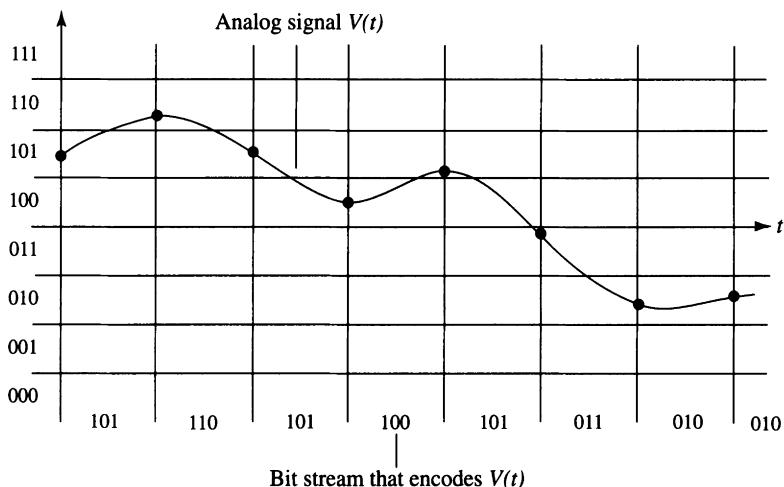
Communication network services worldwide are growing exponentially. This is not a new phenomenon. In affluent countries, the telephone network has for many years been accessible to virtually every business and home. In the U.S. cable TV is reaching one home in two, and a quarter of the population 16 years and older has access to the Internet at work or at home. In developing countries where penetration of the telephone network is significantly below 100%, there is a large unfilled demand, leading to a waiting time for telephone access of several months to years. In this section we describe the four principles that underlie the growth of communication network services: digitization, economies of scale, network externalities, and economies of scope or service integration.

### **1.2.1** Digitization

There are two aspects to digitization. First, any information-bearing signal can be represented by a binary string with an arbitrarily high degree of accuracy (explained later in this section). Second, it is much cheaper to store, copy, manipulate, and transmit a digital signal than an analog signal, because advances in electronics have made digital circuits much more robust and cheaper than analog circuits. Because of these two aspects, the overwhelming majority of today's communication systems are digital.

If a signal is in the form of a sequence of discrete symbols, it is obvious that this signal can also be represented by a binary string. For example, since there are fewer than  $2^7 = 128$  characters on the computer keyboard, we can represent each character as a unique 7-bit sequence. Thus, any character string can be represented as a binary string seven times as long. Conversely, by decoding the binary string, 7 bits at a time, we can recover the character string.

Binary representation of an analog signal, such as voice, requires two steps, illustrated in Figure 1.13. The first step, called *sampling*, consists of measuring periodically the value of the analog signal  $V(t)$ , a real-valued function of continuous time. These values, called *samples*, are represented by the small dots in the figure. The second step, *quantization*, consists of representing the samples by a fixed number of bits. To quantize the samples, the digitization



**FIGURE**  
1.13

Digitization of an analog signal  $V(t)$ . The signal is sampled at a rate  $2 \times f_{max}$  where  $f_{max}$  is the maximum frequency in the signal. The samples are quantized with  $N$  bits per sample. The resulting bit stream has rate  $2N \times f_{max}$ , and the signal-to-noise ratio is  $6N$  dB.

hardware decomposes the range of possible sample values into a finite set of intervals, called *quantization intervals*, and associates a different binary number with each interval. The hardware then represents a given sample by the binary number associated with the quantization interval of the sample. (If there are  $2^n$  quantization intervals, each interval is represented by an  $n$ -bit word. In the figure, the eight intervals are represented by 3-bit words.) Using these two steps, the hardware represents the analog signal  $V(t)$  by the bit stream composed of successive binary numbers associated with the quantization intervals of the samples.

The theoretical basis for sampling is *Nyquist's theorem*. That theorem states that no information is lost by the sampling provided that the sampling rate is at least twice the maximum frequency, or *bandwidth*, of the analog signal. This theorem makes precise the fact that the faster a signal changes—that is, the larger its bandwidth—the more frequently it must be measured to observe its variations.

The quantization of a sample approximates its value by one representative value in the quantization interval. Consequently, quantization introduces errors. The errors are small if the quantization intervals are small, which is the

case if the range of values of the signal is decomposed into a large number of quantization intervals. Equivalently, we can view these errors as the addition of some analog noise to the original signal. That equivalent noise is called the *quantization noise*. We measure the importance of the quantization errors by the ratio of the power of the signal over that of the quantization noise. This signal-to-noise ratio is usually measured in *decibels* (dB). If the ratio is equal to  $R$ , then its value expressed in decibels is  $10 \log R$ , where log designates the logarithm in base 10. For example, 3 dB means a signal power that is twice as large as the noise power, since  $10 \log 2 \approx 3$ . In telephone transmission, a signal-to-noise ratio of about 48 dB is acceptable. In high-fidelity audio applications, a signal-to-noise ratio of 65 dB or better is desired. A low-quality cassette deck, for example, has a signal-to-noise ratio of about 55 dB, where the noise is due mostly to the granularity of the magnetic material on the tape. A high-quality cassette deck has a signal-to-noise ratio of about 68 dB, where the magnetic noise is attenuated by some signal processing such as Dolby C.

The signal-to-noise ratio due to quantization is approximately equal to  $6N$  dB, where  $N$  is the number of bits used to represent each sample. (Thus in Figure 1.13 the ratio is 18 dB.) This result can be explained as follows. If  $N$  bits are used to number the quantization intervals, then there are  $2^N$  such intervals, and a typical error has a magnitude proportional to  $2^{-N}$ . Since the power of the quantization noise is proportional to the square of the magnitude of that noise, we conclude that the noise power is proportional to  $2^{-2N}$ , yielding a signal-to-noise ratio in decibels of about  $10 \log(2^{2N}) = 2N \times 10 \log 2 \approx 6N$ .

If the digitization hardware uses  $N$  bits per sample and samples the signal  $f_s$  times per second, then it produces a bit stream with rate  $N \times f_s$  bps. For example, the telephone network transmits frequencies in the voice signal up to 4 kHz and achieves a signal-to-noise ratio approximately equal to 48 dB. To meet these objectives, the sampling rate must be  $2 \times 4$  kHz = 8 kHz, and the number of bits per sample must be equal to  $48/6 = 8$ . Consequently, the digitized voice signal has a rate of  $8$  kHz  $\times$  8 = 64 Kbps.

In a compact disc, or CD, the target specifications are a maximum frequency of 20 kHz and a signal-to-noise ratio of 96 dB. These specifications require a sampling rate of at least 40 kHz and at least 16 bits per sample. A stereo signal, then, corresponds to a bit stream rate of at least  $1.3 \times 10^6$  bps. Consequently, a 70-minute CD must store at least  $70 \times 60 \times 1.3 \times 10^6 / 8 = 682.5$  MB. This large storage capacity of CDs explains why CD-ROMs are used to distribute digital information.

As a final example, consider a television signal. The NTSC TV signal has a maximum frequency of 4.5 MHz. If the signal-to-noise ratio must exceed 48 dB,

then the bit stream must have a rate of about 72 Mbps. Note that these examples ignore the savings in bit rate that can be accomplished by data compression techniques.

### 1.2.2 Economies of Scale

Communication networks exhibit scale economies. That is, the average cost per user of the network declines as the network increases in size, measured by number of users, subscribers, or host computers. There are several reasons for this declining average cost. First, owing to advances in communications technology, primarily optical communications, the cost of a transmission link grows at a slower rate than does its capacity or speed. Hence, if the bit streams generated by  $n$  users can be made to occupy the same link with a capacity of  $nB$  bits per second at an average cost of  $C(nB)$ , then each user may continue to generate a bit stream at rate  $B$ , but the per user cost  $C(nB)/n$  will decline as  $n$  grows. Note that to take advantage of cheaper high-speed transmission, the low-speed bit streams of individual users must be combined into a high-speed bit stream. This is possible to do with the techniques of multiplexing and switching introduced in section 2.6.

Second, a network has certain fixed costs of operations, administration, and maintenance. Because these costs are not sensitive to network size, the per user share of these fixed costs declines with the number of users. Third, in networks dedicated to distribution services, such as cable TV, where the same bit stream is delivered to many users, the sharing of transmission facilities allows the per user cost to decline with the number of users.

Scale economies leading to declining average cost are present in many industries, including telephone, CATV, electric power transmission, and water distribution. In these industries, a large company enjoys a lower average cost than a smaller company. The large company can lower its price and drive its smaller competitors out of business. (This happened in the United States in the early years of the electric power and telephone industries.) Thus, these industries have a tendency to create one large monopoly, resulting in government regulation of these industries in most countries. In the United States the communications industry is regulated by the Federal Communications Commission (FCC) at the national level and by public utility commissions at the state level. The objective of this regulation is to prevent large companies from reaping monopoly profits or unfairly competing with small companies.

Scale economies may be eroded by further technological advances. Today's small gas-turbined power plants are as efficient as yesterday's large plants. The smaller plants can be more flexibly located, and investing in them is less

risky. So throughout the world there is a movement to deregulate the electricity generation business. The power transmission grid remains a monopoly, but regulators are insisting on giving small power producers "equal access" to the grid so they can compete with the larger producers.

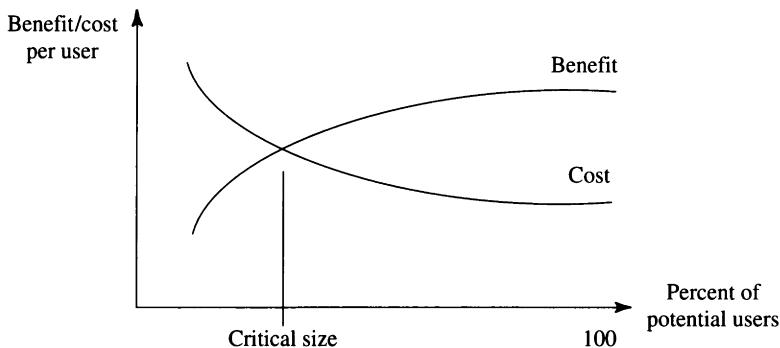
Something similar has happened in telecommunications. Small-scale long-distance communication networks and data networks are efficient and, as a result of guaranteed "equal access," there are many long-distance telephone carriers and more than 5,000 Internet Service Providers (ISPs) in the U.S.

### 1.2.3 Network Externalities

A network service is said to have positive externalities if its value to a user increases with the number of users. The clearest example of externalities is the case of telephone service: the value to a telephone subscriber increases as more people subscribe, because the subscriber can then talk to more people. A less obvious example is the formation of special interest groups of users of a data network. These groups provide services to their members, such as operating an electronic bulletin board dedicated to a special topic of interest. The presence of network externalities can be inferred from the fact that the group must become sufficiently large before it is viable.

Externalities provide a powerful incentive for internetworking. When two independent networks are interconnected, the value to the users of both networks increases. The extra resources needed to implement the interconnection are few if the two networks follow compatible standards. The combination of network scale economies, which reduce per user cost as the network grows, and network externalities, which increase per user benefits as the network grows, creates positive feedback that can lead to an exponential growth in demand and supply of network services. Examples of this growth occurred in the demand for facsimile machines and Internet access.

This growth phenomenon is illustrated in Figure 1.14. If the number of users is below a critical size, then the per user cost exceeds the benefit, and users would not be willing to pay for that service. Once the critical size is exceeded, users are increasingly willing to bear the cost. For example, the French telephone company gave free terminals to its subscribers for use in accessing a variety of private on-line services such as news, train schedules, and restaurant menus on the Minitel network. The minimal device for accessing the network is a dumb terminal with a monochrome display and dual mode 75/1200-baud modem (PCs can also be used). Within a short time a very large set of services was offered, and the telephone company quickly recovered the cost of the terminal through greater use of the telephone network.



1.14

FIGURE

As more users subscribe to a network, the cost per user decreases, and the benefit per user increases. This positive feedback further fuels the growth of the network.

A counterexample is provided by AT&T, who in the 1960s developed a video telephone called Picturephone. The product was a commercial failure because the number of subscribers remained below the critical size.

Figure 1.14 makes it clear that to initiate a new network service there must be an initial period in which users are subsidized. The subsidy, which may come from the government or from a company's shareholders, lowers the cost that users face, and hence the critical size. For example, the ARPANET computer network was paid for entirely by the U.S. government. It later developed into the Internet, which now needs no subsidy. The initial subsidy is often provided by users of existing services, through service integration.

### 1.2.4 Service Integration

Economies of scope, or service integration, refers to the fact that a network that currently provides one set of services may be expanded to provide new services at an additional cost that is much less than if a separate network were built to provide those new services. Economies of service integration are possible because communications engineers now design services in a modular and standardized way so that new services can be introduced using existing hardware and software modules.

The widespread deployment of ATM, described in Chapter 6, and broadband access over cable TV and ADSL, described in Chapter 5, will facilitate service integration to such an extent that one can imagine a single network that will provide all of the services that today are provided by separate networks.

These services include telephone, data, broadcast TV and radio, and CATV. Information carried by newspapers, magazines, books, and other forms of print media could also be provided over this network. The potential economies and profits are enormous. This potential explains the current struggle among telephone, computer, TV, and entertainment companies to form coalitions that will own and operate this “universal” network. Wireless networks offer users the freedom of mobility. If they are able to integrate voice and data, the demand for wireless services may grow even more.

## **1.3**

## **FUTURE NETWORKS**

In this section we compare the capability of the Internet, ATM, and cable TV networks to provide the core technologies that can be used to develop the “information superhighway.” We use this popular term to designate a high-performance, flexible communication network that can provide all of the services now provided by the three types of networks. We assume that wireless will continue for some time mainly to serve as a means of access to the wireline network.

### **1.3.1**

### **The Internet**

The Internet today comprises hundreds of thousands of local area networks (LANs) worldwide, interconnected by a backbone wide area network (WAN). LANs typically operate at rates of 10 to 100 Mbps. Until 1995 links of the WAN supported lower bit rates, but the dramatic increase in traffic combined with the reduction in the cost of optical links have increased backbone link rates to as much as 10 Gbps. To deal with these large link rates, some network service providers deploy IP routers interconnected by ATM switched networks.

Corporate data networks typically have a much smaller speed connection to the Internet backbone compared with their LAN speeds. This is appropriate because LANs support the high bit rate traffic between workstations and file servers within a single organization. The WAN, on the other hand, supports electronic mail and infrequent file transfers that can be accomplished with low-speed connections. For example, in 1995 40,000 students, faculty, and staff at the University of California, Berkeley, had access to the Internet using 20,000 workstations and PCs. Within the university campus these computers were interconnected by 10-Mbps Ethernets and 100-Mbps FDDI rings. The Internet traffic between the campus and the rest of the world was handled by two 1.5-Mbps links, with an average utilization of 30%. By comparison, the telephone

links between the campus and the rest of the world have a capacity of 200 Mbps.

Users access the Internet in one of two ways. Within a large company, government agency, or university, the user's PC or workstation is attached to a LAN that is part of the Internet. Users at home and in small companies subscribe to an ISP. Subscribers use low-speed modems to connect their PCs to ISP hosts, which, in turn, have Internet access. Some users are changing to higher speed access through cable TV or ADSL.

We consider here some of the many factors that have contributed to the spectacular success of the Internet, which by 1999, at 25 years of age, connected 300 million users worldwide. For users connected to LANs the incremental cost of Internet access is a small fraction of the cost of the LANs. This insignificant incremental cost, combined with the network externalities of services like e-mail, Web access, and formation of special interest groups, led to an exponential growth. Second, more and more people own PCs, and newer PCs come equipped with networking hardware, including a built-in modem, and software. For these users, the cost of Internet access is only the charges of their ISP provider. In 1999 ISPs provide unlimited access at 28.8 Kbps for an affordable \$20 monthly charge. Higher speed access over cable TV or ADSL costs \$40 per month. The combination of positive networking externalities and the steady reduction in the cost of computers accounts for some of the Internet's exponential growth.

Additional growth is fueled by innovative, low bit rate, delay-insensitive applications such as the World Wide Web (WWW), with icon-driven interfaces that make browsing easy. Internet applications such as e-mail and file transfer can be provided at a cost that no alternative network can match. Lastly, designers of these new applications often distribute them freely. They do so because the Internet has been developed by, and in turn has helped to sustain, a remarkable cadre of experts who strongly support keeping the Internet a free and open network. The successful introduction of commercial software such as WWW browsers may also require an initially subsidized distribution of the software to overcome the critical size depicted in Figure 1.14. These issues are discussed in Chapter 10.

One future development of the Internet, then, is more growth of the same kind: more users and more low bit rate, delay-insensitive applications for which the Internet has an overwhelming cost advantage.

Another possibility is that the Internet will develop into the information superhighway by supporting real-time, high bit rate, delay-sensitive applications such as interactive voice and video applications. To support those applications, the Internet will need to change in three ways. The backbone links will

have to be upgraded, and the network switches for those links must be replaced by switches with very large throughput and low delays. This change is well on its way. At that point network designers may replace the IPv4 (Internet Protocol version 4) network layer, which cannot guarantee the delay, bandwidth, and loss bounds that real-time applications need. IPv4 could be replaced with a newer version, IPv6, or with ATM, or more likely, with enhancements to IPv4 that meet some of those needs. (The Internet Protocol is discussed in Chapter 4.) Recent commercial routers support some rudimentary resource reservation needed for real-time applications. The decision to migrate to ATM for the high-speed, low-delay links of the Internet may be forced by the advantages of ATM over IP. However, at present, applications and operating systems that exploit this native ATM capability are rare. These latter considerations point to an Internet growth path built on ATM technology and high-capacity links.

### 1.3.2 Pure ATM Network

ATM technology is capable of carrying a wide range of information transfers, from e-mail to videoconferences, with a range of quality of service that can match user needs. Thus, ATM could provide the full range of services contemplated as offerings on the information superhighway.

To implement such a network, network service providers would install high-speed ATM access lines to users over optical-fiber or ADSL subscriber loops. The ATM cells would be switched by high-speed ATM switches in the backbone network. The result would be a “universal” network, built on sound theoretical principles.

Several obstacles to the realization of this scenario have been overcome in the past five years. First, the cost of providing high speed (up to 1.5 Mbps) over cable TV or the telephone loop has come down. Second, large, high-speed ATM switches that allow the network to provide the wide range of services are coming to the market (although ATM interfaces for user equipment still remain expensive). The most significant challenge for ATM, however, is to satisfactorily serve the enormous legacy IP infrastructure and the continuing advances of IP that will meet at least some quality of service needs.

### 1.3.3 Cable TV

As explained in section 1.1.3, the CATV industry is developing a bidirectional network that can deliver video programs controlled by users. The user interface rate of such a network is orders of magnitude larger than in the telephone

network. The increased rate comes from using fiber-to-the-curb plus local coaxial networks instead of twisted pair subscriber loops.

By using frequency-division multiplexing, this CATV network can also provide data, telephone, and compressed videophone services. Connecting this CATV network to the telephone and other wide area networks would allow a wide area multimedia network to be implemented. On such a network, information could be transported locally by the CATV network and, over long distances, by ATM over the SONET network of the telephone company. This fusion of the CATV and telephone networks will be facilitated by collaborations, perhaps of the kind indicated by AT&T's acquisition of TCI and MediaOne. A similar outcome could result if the phone company upgrades its low-speed copper subscriber loop network with a high-speed fiber-to-the-curb plus local coaxial network. Such a plan was seriously considered five years ago by some phone companies, but it seems to have been shelved.

### 1.3.4 Wireless

Unlike the Internet and the wireline telephone network, wireless networks in the U.S. are fragmented by the absence of a single standard. Wireless phones conforming to one standard do not work in another region served by a company operating with another standard. So wireless telephony in practice provides mobility only in a local region. This situation is likely to persist for some time, until the emergence of companies that stitch together local markets to provide national coverage.

Greater uncertainty surrounds the use of notebook and palm-sized wireless devices for access to the Internet or corporate intranets. Such devices offer the advantage of mobility and do not require wiring. However, many advances in devices and networking are needed (see discussion in Chapter 7) before wireless Internet access becomes significant.

### 1.3.5 And the Winner Is . . .

We can be sure that no single network technology will emerge as the undisputed winner. The reason is in part technological, in part economic. The technological reason is that all three different technologies (ATM, Internet, CATV) are converging to provide an overlapping set of services. Thus, to a limited extent each technology can substitute for the others. The economic reason is that the large investments being made in all three types of networks mean that they will all be deployed for a long time. The 1996 Telecommunications Act eliminates many ownership and regulatory barriers in long-distance, local telephone, and cable TV markets. The Act will spur mergers of companies and

intensify competition, especially between local and long-distance telephone companies. Thus the information superhighway will be characterized by a collection of heterogeneous networks, offering a variety of services.

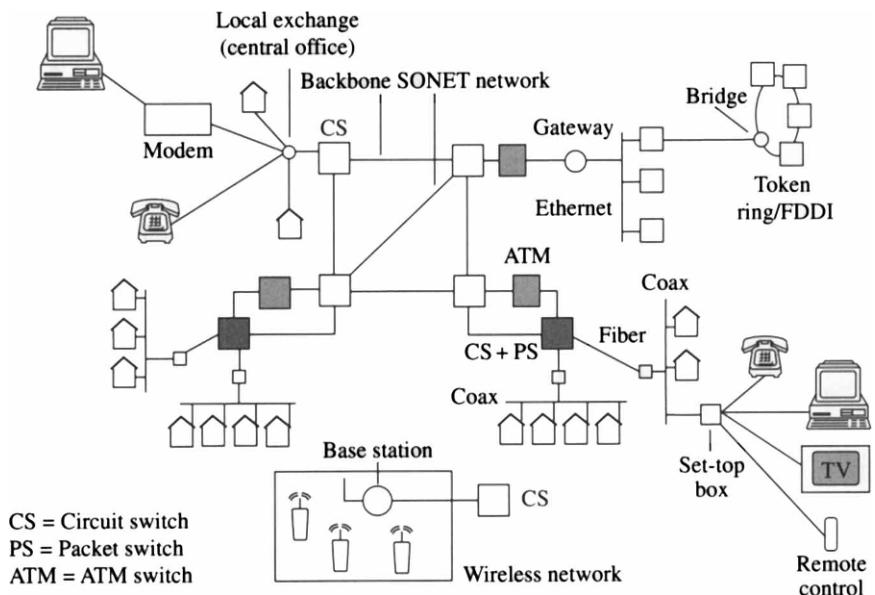
Within five years we can expect to see the wide deployment of a large number of video and data services, such as movies, video on demand, teleshopping, and Internet access, over hybrid fiber/coax cable networks. Those networks will be deployed by telephone companies in collaboration with CATV companies. Once this network is in place, the communication cost of distributing video programs to a large number of users will be reduced considerably, and profits will accrue to those companies that can provide video services or other content that people are willing to purchase.

The Internet will grow using IP with faster links and routers and new protocols that better control the quality of service. Growth will come from the interconnection of more and more LANs and from the introduction of new services. Internet protocols are evolving to include resource reservation and call admission and may well eventually incorporate a pricing component to regulate the reservation of resources. Thus, the Internet will continue to provide the best-effort services, such as e-mail, file transfer, and WWW, for which it was initially designed. The potential of such applications is enormous, as demonstrated by the rapid growth of on-line business-to-business commerce. The Internet will increasingly attempt to capture the potentially large market of real-time traffic, including phone service. The capability of the Internet to carry real-time traffic will increase as high-speed ATM networks become extensive in the Internet backbone network.

The demand for ATM is being expressed through ATM switches that interconnect Ethernet LANs. ATM WANs are becoming deployed because they enable very fast switching and permit quality guarantees to accommodate data, telephone, and video services in the same network.

Finally, there will be a large increase in mobile and wireless access to networks. Terminals resembling today's notebook computers, connected to base stations by wireless radio or to wireless LANs, will become common.

Figure 1.15 sketches a version of the information superhighway with these features. It includes a backbone circuit-switched SONET network with links at speeds of gigabits per second. The SONET network provides transport for ATM services, as well as circuit-switched connections for carrying video programs to local CATV head stations. The latter distribute those programs over a fiber-coaxial distribution network. Control messages from users are sent over a packet-switched network. ATM services are used to provide wide area transport for Internet traffic generated over FDDI or Ethernet local area networks. Lastly, wireless terminals access base stations connected to the wired networks.



1.15

FIGURE

The future network will be heterogeneous, scalable, and flexible.

The challenge is to interconnect these networks in ways that accommodate this heterogeneity, that are extensible, and that provide the range of quality of service needed to support a large variety of information services. In the rest of this book we will be concerned with defining these challenges and offering plausible responses.

## 1.4

## SUMMARY

Communications services today are provided by wireline and wireless telephone, CATV, and data networks. These networks serve different markets using different technologies. However, technological advances are dissolving those differences, and these networks are converging in their ability to provide integrated services. The drive toward convergence is spurred by digitization, economies of scale, networking externalities, and economies of scope. ATM represents one major technological integration of these developments. The future information superhighway will comprise a collection of heterogeneous networks: circuit-switched SONET networks that carry telephone and video traffic and provide for the transfer of ATM cells, fiber-coaxial cable distribution

networks for video, wireless access, and a ramified Internet that provides the most economical best-effort packet transport and modified protocols to control the quality of service. The challenge to network engineers is to make it possible to interoperate these heterogeneous networks in ways that are extensible and secure and that provide the necessary range of service quality.

## 1.5

---

## NOTES

An excellent Web site with 7000 links to telecom information resources on the Internet can be reached at <http://china.si.umich.edu/telecom/telecom-info.html>.

International Telecommunication Union (ITU) publications may be viewed or purchased through <http://www.itu.int>.

An informative review of global trends in the telecommunications industry can be found at [http://www.bt.com/global\\_reports/](http://www.bt.com/global_reports/).

A brief history of the Bell System with an extensive account of the economic and political factors that led to the deregulation of the long-distance telephone industry in the United States is given in [TG87]. A description of the ARPANET philosophy is available in [Cl88]. The current cable TV technology is described in [C90]. The applicability of ADSL for video dial tone is studied in [CW94]. Technical discussions of high-speed digital subscriber loop technology, including HDSL and ADSL, are presented in [JSAC95]. There is a large volume of literature on digital signal processing; [GG92] is devoted to compression techniques. There was a flurry of papers in the 1970s on the economics of networking externalities of which a typical example is [AA73]. The example of Minitel is reviewed and compared with the Internet in [LLKS95]. Many popular books on the Internet have appeared recently. There are no reliable figures on its size or growth (measured by number of hosts, subnets, users, activity), how it is used, its cost, the extent of congestion, and so forth. A summary of a survey of Internet users is available on the Web [CN95]. ATM is described in Chapter 6. A 1995 issue of *IEEE Network* is devoted to video dial-tone networks [IN95].

## 1.6

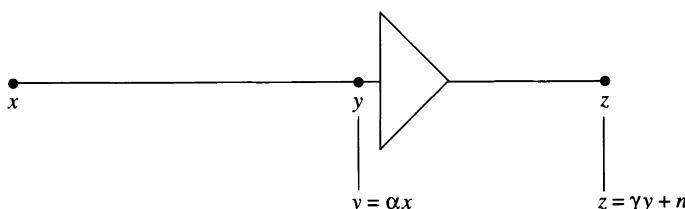
---

## PROBLEMS

1. Using modems to exchange information can be inefficient. Suppose you use a 9.6-Kbps modem to type an e-mail message from your home computer to a friend's computer. During this transmission, a telephone connection is

established. If you are a fast typist, you can type 80 words (400 characters) per minute. Show that you are using less than 0.5% of capacity of the modem connection. By first typing your mail into a file and then transmitting it, you can send it at the full line rate.

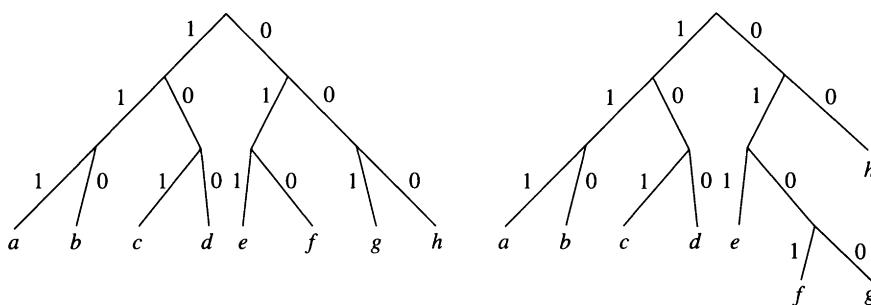
2. In a packet-switched network, the transmission line is shared by many users. How many users' e-mail needs can be accommodated by a shared 9.6-Kbps line? Assume that the average e-mail message consists of text composed by typing and copying an existing file in the proportion of 1 to 5.
3. Consider a communication link that transmits 10,000 bits per second. The objective is to transmit a file of  $\$B\$$  bits. A synchronous transmission is used. The bits are sent in packets of  $\$P\$$  bits. Each packet contains 16 extra bits, which are used for error control. Two packets must be separated by at least 10ms. Find the total time taken to transmit the file as a function of  $\$B\$$  and  $\$P\$$ .
4. We develop a model to estimate the feasible length of a CATV distribution network. The network consists of a series of sections. Each section comprises a length of cable, which attenuates the signal, followed by an amplifier, which boosts the signal strength but also adds noise. If the input power to the cable is  $x$ , the output power is  $y = \alpha x$ , where  $0 < \alpha < 1$  ( $\alpha$  depends on the length of the cable). If the input power to the amplifier is  $y$ , its output power is  $z = \gamma y + n$  where  $\gamma > 1$  is the gain and  $n$  is noise (refer to Figure 1.16). Suppose the power of the CATV signal at the beginning of the distribution network is  $S$ , and suppose the signal goes through a network that is  $K$  sections long. At the end of this network the signal  $Z$  is a sum of two terms,  $Z = X + N$ , where  $X$  is the CATV signal power and  $N$  is the noise power. For proper reception, we need (1) minimum CATV signal strength  $X \geq X_0$  and (2) a minimum signal-to-noise ratio  $X/N \geq R_0$ . How



1.16

**FIGURE**

A section of a CATV distribution network consists of a length of cable, which attenuates the signal, followed by an amplifier, which boosts the input signal and adds noise.



1.17

FIGURE

A finite alphabet is encoded into binary words by associating each letter with a leaf of a binary tree. The tree on the left gives a fixed-length encoding; the tree on the right gives a variable-length encoding.

long can the network be? We can increase the network length by increasing the input signal power  $S$ . We can also replace the cable by an optical fiber whose  $\alpha$  is much larger than that of cable, but for which the input signal level  $S$  is much smaller. Discuss the trade-off between increasing  $S$  and increasing  $\alpha$ .

5. Suppose sequences of letters from a finite alphabet are to be transmitted over a binary communication channel. Assume that there are  $2^n$  letters in the alphabet, so that each letter can be encoded into an  $n$ -bit word, as in the left panel of Figure 1.17. An  $m$ -letter sequence is thus encoded into a binary sequence of length  $mn$ . A variable-length encoding is obtained, as in the right panel, by associating each letter with a leaf of the tree. Thus the letters  $f$  and  $g$  are encoded into 4-bit words,  $h$  is encoded into a 2-bit word, and the rest into 3-bit words. A variable-length encoding is preferred if some of the letters are used more frequently than others. Suppose, in the example above, the letter  $h$  is used with probability 0.5, and the remaining seven letters are used with probability  $1/14$ . Show that the average length of the binary encoding on the right is smaller than the average length on the left. If there are  $N$  letters in the alphabet and they occur with probabilities  $p(1), \dots, p(N)$ , then the minimal length encoding requires  $H$  bits per letter, where  $H$  is the entropy:

$$H = - \sum_k p(k) \log_2 p(k).$$

Show that if all letters are equally likely, the minimal length encoding requires  $\log_2 N$  bits per letter.

6. A customized CD company might operate as follows. All its music would be available by means of central servers. Customers would come into any branch store and select the songs they want. Bit streams representing those songs would be transported over the network from the servers and written onto a CD. The company would save money since it would have no inventory. Is the scheme feasible? As we have seen, a 70-minute CD stores about 700 MB. How large a communication bandwidth should be provided between server and store so that a customer would not need to wait for more than 10 minutes? If a 45-Mbps link has a monthly rent of \$20,000, and a CD sells for \$20, and if a 100% markup is needed for other costs, is this a profitable idea? How much should the rent be to make the scheme profitable? Can you think of other business opportunities where communication substitutes for inventory?
7. Sketch the data network in your campus or company. How many hosts are there, and how large is the user population? What is the speed of the access link to the Internet? How do you gain access to the Internet? How much does home access to the Internet cost?
8. Sketch how your telephone is connected to the central office of your telephone company. How large is the switch located in that central office? How can you gain access to different long-distance phone companies using the same local telephone company?
9. Sketch the CATV network in your city. It is likely that the city has granted this network a franchise, meaning that it has the exclusive right to operate in your city. What arguments can you muster for and against such a franchise?
10. Besides CATV and broadcast TV, what other means do you have to receive video programs? How would you characterize the differences between these alternative sources of supply from the customer's point of view?
11. Is there a newspaper in your area that you can read on-line, for example on the World Wide Web? If there is, compare that service with the printed version of the same newspaper.
12. Recent purchases of CATV companies suggest that they are valued at \$3,000 per customer, so that if a CATV company has one million customers, its market value is about \$3 billion. To obtain such a market value, the CATV company must make profits of about \$300 per customer per year. How does the company make such profits? Look at the annual reports of some CATV companies in your library.

13. What is the revenue per residential and business customer of your local phone company?
14. Compare two proposals to distribute movies. Assume that in both schemes 100 video channels are available to distribute movies to 300 households. In the first scheme, which uses the existing CATV network, the CATV operator finds out the 25 most popular movies and plays them continuously starting every 15 minutes. In the second scheme, households order the movies they want by using a control channel. This scheme requires costly modification of the existing CATV network to create the control channel and additional terminal equipment that users must employ to register their demand.

Households must pay for watching a movie. Build a model for household demand that can be used to predict which scheme will make the greater profit. How would you go about collecting data to estimate or validate your model? What is your guess as to which scheme is more likely to be profitable and why?

15. One common approach to public regulation of a monopolistic telephone company is called *rate of return regulation*, in which a company's profit is limited to  $\rho \times V$ , where  $\rho$  is a "fair" rate of return (say, 15%) and  $V$  is the value of the company's assets. ( $V$  can be measured as the depreciated value of the company's past investments.) It has been suggested that rate of return regulation encourages the telephone companies to invest in more capital equipment and hire fewer workers than is optimum from society's viewpoint. Can you argue in favor of or in opposition to this suggestion?

# Network Services and Layered Architectures

**M**any networks provide transportation services. The postal system, whose services include the transfer of letters and parcels, is a familiar example. The postal system's services are differentiated by quality: there is registered mail, overnight delivery, surface mail, third-class mail. These services are built from basic transportation services, such as truck or rail or air transport. The postal system uses these basic services to create the more sophisticated services that its customers purchase. If you mail a letter, the system selects a route over which to send it; puts the letter with others going on the same route in a larger container; ships the container using air transport, say; transfers the container to a post office near the destination using truck transport; and finally brings the letter to the destination using yet another service, namely hand delivery by the letter carrier.

Characteristics of the basic services and system performance are summarized and measured by a few parameters: the volume of letters that can be handled per day or per hour, the speed of delivery, the fraction of letters that are lost. To a considerable extent these characteristics are determined by the capabilities of the postal system "hardware": the number of trucks, train cars, and airplanes the postal system has; their speeds; the routes they can use; and so on. These hardware-determined capabilities are then managed and controlled by the postal system's "intelligence"—embodied in hardware or software systems or postal system employees—to produce the more sophisticated services offered to customers. Naturally, the characteristics of the underlying basic services limit the range and quality of the sophisticated services that can be provided. For example, if the postal service used only railway and truck

transport, it would not be able to offer overnight delivery. Over time, postal services of different countries have been *interconnected*. Lastly, through service integration, the postal system uses its existing resources to offer new services: you can "wire" money and telegrams, you can mail a letter and send a fax, you can open a post office box and a postal savings account, and so on.

We will find that the key concepts introduced in this simple description of the postal system carry over with appropriate reinterpretation to the case of communication networks. Those concepts are user services and service quality, basic or bearer services and their characteristics, the underlying "hardware," and management "intelligence." We have already encountered the concepts of network interconnection and service integration in section 1.2.4.

The services provided by communication networks enable users to exchange information. There is a wide range of services: users can talk to each other over a telephone network, transfer data over a computer network, and watch a video program over a cable TV network. Since users interact with the network through some terminal device (telephone, computer, TV controller), it is sometimes more appropriate to say that the network services are used (consumed) by user applications (processes running on the terminal device).

Engineers build a network by interconnecting two types of "hardware" or network elements: transmission links and switches. Links transfer strings of bits from one location to another. Switches are computers that store, route, and manipulate those bit strings. This hardware supports the network's *bearer services*: the transfer of bit strings, in a few standard formats, from one source or user to one or more network destinations. Bearer service performance characteristics are summarized in a few parameters: the acceptable formats; the connectivity and selection of routes from source to destination; and the speed, delay, errors, and so forth, of the bit string.

A network can effectively support a particular user application only if its bearer services have the requisite characteristics. To support voice conversations, for example, the end-to-end delay should not be more than 200 milliseconds (ms), say. To support data transfer the error rate should not be more than  $10^{-4}$ , say, and so on. The most demanding applications, such as the transfer of X rays with a fidelity and speed that make them acceptable for use by radiologists for diagnosis and interactive videoconferencing, require a high-performance network.

More sophisticated services are built from less sophisticated ones in a layered hierarchy or architecture. Each layer adds functionality. Each layer comprises specific rules of control and management, implemented in software or hardware. These rules and the software that implements them are called *protocols*. The protocols reside in the switches and host computers. Some sets

of protocols have become standardized, and interconnection of networks is facilitated if they conform to the same standard.

This chapter introduces the concepts of how networks function and the logical layers used to divide networks into smaller subsets of functionality. By the end of this chapter you will understand the mechanisms used to implement network functions, including error-control schemes such as the Alternating Bit and Go Back N protocols and flow and congestion control.

We start by discussing a few commonly used applications in section 2.1. In section 2.2 we examine the communication traffic generated by user applications. We comment on the information that different types of user applications exchange. In particular, we explain why different applications require network services with different characteristics.

In section 2.3 we describe network services and discuss their characteristics. These characteristics match those required by user applications.

In section 2.4 we identify high-performance networks in terms of their scalability, their ability to support demanding and diverse user applications, and the ease with which they can be connected to other networks.

In section 2.5 we discuss network elements and explain how the properties of these elements affect the characteristics of the services that they implement.

We explain the mechanisms that the network elements use to implement services in section 2.6.

In section 2.7 we introduce the layered organization of network operations. We explain that the main advantages of a layered organization are modularity and standardization.

In section 2.8 we present the Open Data Network (ODN) model. This model provides a useful framework for describing how networks are organized. The Open Systems Interconnection or OSI reference model, used in data networks, is presented in Chapter 3.

Section 2.9 discusses how the various components that we explained in the chapter are assembled in common network architectures.

Section 2.10 discusses the network bottlenecks and examines the technological improvements that are required to improve the networks.

## 2.1

---

## APPLICATIONS

To make the discussion of information transfers more concrete, we start by discussing a few commonly used network applications. We explain how these applications work and describe some of their characteristics.

### 2.1.1 World Wide Web

The World Wide Web is a distributed application that enables you to navigate through a set of hyperlinked documents, called *Web pages*. Each Web page may contain text, pictures, audio clips, video clips, and possibly links. A link specifies either a location in the same document or another Web page location and name. The location may be another file in the same computer or in another computer attached to the Internet. When you click on a link, the application arranges to display the new location in the document or to transfer and display the new Web page. Accordingly, when you browse the Web, you initiate a sequence of file transfers. The size of a Web page typically ranges from a few kilobytes to a few hundred kilobytes. If the link is to a video clip or to a large file, then the transfer may be of a few megabytes. As you may have experienced, some Web pages take a long time to appear. This delay is not difficult to understand. For instance, if the requested Web page corresponds to 100 KB and if the transfer rate is limited to 8 Kbps, then the transfer takes about two minutes. The transfer rate is limited not only by your modem but also by other connections that share some critical links of the network with your own transfer. We expect Web page transfers to be error-free. Accordingly, transmission errors should be corrected.

### 2.1.2 Audio or Video Streams

Streaming audio and video applications enable you to listen to or view a program as it is being transferred. Many radio stations transmit "live" on the Internet, and some live feeds are also available from TV stations. These streaming applications generate streams of packets that the network delivers from the source to the destination. During their travel across the network, the packets suffer variable delays and some packets get dropped. The destination buffers a few packets before it starts "playing" them back as an audio or video stream. The buffering absorbs the delay fluctuations. Note that, with the buffering, all the packets face a delay that is at least equal to the maximum value of the delay across the network. Indeed, to play back the packets at the same constant intervals as they enter the network, all the packets must have the same total delay. Consequently, the faster packets must be delayed so that they have the same delay as the slowest one. Since these applications are one-way transmissions, the fixed delay is unimportant. The transmission rate of such applications depends on the quality of the program. The rate of an audio transmission typically ranges from 8 Kbps to 30 Kbps. A video transmission has

a rate between 40 Kbps and 80 Kbps. Some transmission errors are acceptable for audio and video. Such errors are perceived as noise or corruption of the pictures.

### 2.1.3 Voice over Packets and Videoconferences

Inexpensive video cameras and audio devices are available to set up telephone calls or videoconferences between PCs. For conversations, the one-way delay is barely noticeable if it is less than 100 ms. Beyond 350 ms, the delay makes the conversation unpleasant. The small delay requirement of voice implies that the voice samples must be placed in small packets. To appreciate this implication, assume that the voice is encoded into a 64 Kbps bit stream. Say that we place the voice bits into packets of 1,600 bits. It takes  $1,600/64,000 = 25$  ms to collect the bits that fill up a packet. Consequently, the packetization introduces a 25-ms delay that adds to the maximum delay across the network. Distortions caused by transmission errors are preferable to the excessive delays that would be required to correct the errors by retransmitting the erroneous packets.

### 2.1.4 Networked Games

Many networked games are played across the Internet, such as board and card games and faster action games. When playing such a game, computers exchange short commands. The acceptable delays depend on the game. The required reliability of the transfer also depends on the game.

### 2.1.5 Client/Server

Many networked applications, from databases to distributed calendars to shared file servers, are organized according to a client/server model. In a client/server application, a server is designed to answer queries from clients. Typically, the client sends a query to the server and waits for the reply. When the reply arrives, the client resumes the execution of its program. The client must be able to detect and react to server or network failures. A server must be able to handle requests that arrive from many clients. A common procedure for meeting this objective is to make the server *stateless*. This term means that the server should not have to remember any information about previous

queries. In practice, the server is rarely stateless. However, application designers attempt to limit the amount of information that the server must remember. The network requirements depend on the acceptable response time of the application.

## 2.2

## TRAFFIC CHARACTERIZATION AND QUALITY OF SERVICE

As the examples in the previous section show, users exchange information through applications. In this section, we examine the characteristics of the information transfers of different applications. These characteristics describe the traffic that the applications generate as well as the acceptable delays and losses by the network in delivering that traffic.

The information that applications generate can take many forms: text, voice, audio, data, graphics, pictures, animations, and videos. Moreover, the information transfer may be one-way, two-way, broadcast, or multipoint.

The information exchanged can be analog or digital. CATV networks, for example, deliver analog video signals to television sets. The telephone network transmits analog or digital voice signals between telephones. Computer networks transfer bit files or bit streams representing text, data, still images, and audio or video signals. Most networks transmit analog signals by first converting them into bit streams, as we explained in section 1.2.1.

In this chapter we limit discussion to digital transmission of information, so user applications eventually require the communication network to transmit bit files or bit streams. We call these bit files or bit streams the *traffic* generated by the application. In order to support a user application, the network must be able to transport in a satisfactory manner the traffic that the application generates. Table 2.1 presents some characteristics about the traffic generated by common forms of information.

Notice that the bit streams generated by a video signal can vary greatly depending on the compression scheme used. When a page of text is encoded as a string of ASCII characters, it produces only a 2-Kilobyte (KB) string; when that page is digitized into pixels and compressed as in facsimile, it produces a 50-KB string. The LaTeX file for this book, including figures, compresses into a 1-MB file. A high-quality digitization of a color picture (similar quality to a good color laser printer) generates a 33.5-MB string; a low-quality digitization of a black-and-white picture generates only a 0.5-MB string.

Information form	Traffic type	Size or Rate
Voice	CBR	64 Kbps
Video	CBR	64 Kbps, 1.5 Mbps
	VBR	Mean 6 Mbps, peak 24 Mbps
Text	ASCII	2 KB/page
	Fax	50 KB/page
Picture	600 dots/in, 256 colors, 8.5 × 11 in	33.5 MB
	70 dots/in, b/w, 8.5 × 11 in	0.5 MB

2.1  
**TABLE**

Characteristics of traffic for some common forms of information.

We classify all traffic into three types. A user application can generate a constant bit rate (CBR) stream, a variable bit rate (VBR) stream, or a sequence of messages with different temporal characteristics. We briefly describe each type of traffic, and then consider some examples.

### 2.2.1 Constant Bit Rate

To transmit a voice signal, the telephone network equipment first converts it into a stream of bits with a constant rate of 64 Kbps (see section 1.2.1). Some video-compression standards convert a video signal into a bit stream with a constant bit rate (CBR). For instance, MPEG1 is a standard for compressing video into a constant bit rate stream. The rate of the compressed bit stream depends on the parameters selected for the compression algorithm, such as the size of the video window, the number of frames per second, and the number of quantization levels. MPEG1 produces a poor quality video at 1.15 Mbps and a good quality at 3 Mbps.

Voice signals have a rate that ranges from about 4 Kbps when heavily compressed and low quality to 64 Kbps. Audio signals range in rate from 8 Kbps to about 1.3 Mbps for CD quality.

For the voice or video application to be of an acceptable quality, the network must transmit the bit stream with a short delay and corrupt at most a small fraction of the bits. (This fraction is called the *bit error rate* or *BER*.)

The end-to-end delay should be less than 200 ms for real-time video and voice conversations, since people find larger delay uncomfortable. That delay can be a few seconds for non-real-time interactive applications such as

interactive video and information on demand. The delay is not critical for non-interactive applications such as distribution of video or audio programs.

The maximum acceptable BER is about  $10^{-4}$  for audio and video transmission, in the absence of compression. When an audio and video signal is compressed, however, an error in the compressed signal will cause a sequence of errors in the uncompressed signal. Therefore, the tolerable error rate for transmission of compressed signals is much less than  $10^{-4}$ .

### 2.2.2 Variable Bit Rate

Some signal-compression techniques convert a signal into a bit stream that has a variable bit rate (VBR). For instance, MPEG2 is a family of standards for such variable bit rate compression of video signals. The bit rate is larger when the scenes of the compressed movie are fast moving than when they are slow moving. DBS (Direct Broadcast Satellite) uses MPEG2 with an average rate of 4 Mbps.

To specify the characteristics of a VBR stream, the network engineer specifies the average bit rate and a description of the fluctuations of that bit rate. We study such descriptions in Chapter 6.

The acceptable delay and BER of these applications are similar to those of CBR applications.

### 2.2.3 Messages

Many user applications on a network are implemented by processes that exchange messages. (For present purposes, a *message* is a variable-length bit string.) For instance, when browsing the Web, a user sends a server requests for Web pages, audio or video clips, or documents. The server replies by sending the requested records to the user. As another example, a distributed computing application generates remote procedure calls for remote machines, which then return the results of the execution of the procedures.

The message traffic generated by various user applications can have a wide range of characteristics. Some applications, such as e-mail, generate isolated messages. Other applications, such as a distributed computation, generate long streams of messages. The rate of messages can vary greatly across applications and devices.

To describe the amount of traffic generated by an application that produces a stream of messages, the network engineer may specify the average traffic rate and some measure of the fluctuations of that rate, in a way similar to the case of a VBR specification.

The network must transfer the messages with an acceptable delay, and it can corrupt only a small fraction of the messages. Typical acceptable values of delays are 200 ms for real-time applications, a few seconds for interactive services, and many seconds for noninteractive services such as e-mail. The acceptable fraction of messages that can be corrupted ranges from  $10^{-8}$  for data transmissions to much larger values for noncritical applications such as junk mail distribution.

Among applications that exchange sequences of messages, we can distinguish those applications that expect the messages to reach the destination in the correct order and those that do not care about the order.

#### 2.2.4 Other Requirements

In this book we will consider only the delay and loss requirements that applications impose on the network. We should remember, however, that other requirements may be important. We mention here reliability and security.

When one or more links or switches fail, the network may be unable to provide a connection between source and destination until those failures are repaired. *Reliability* refers to the frequency and duration of such failures. Some applications (e.g., control of electric power plants, hospital life support systems, critical banking operations) demand extremely reliable network operation. Typically, we want to be able to provide higher reliability between a few designated source-destination pairs. Higher reliability is achieved by providing multiple disjoint routes between the designated node pairs.

Recall that in a multiple-access network such as Ethernet, every computer "hears" every packet that is transmitted. The case of wireless phone transmission is similar. In these networks, to guarantee privacy of transmissions, it will be necessary to encrypt those transmissions. More generally, *security* is concerned with measures that can be taken to prevent unauthorized access to data or information transfer. As money and other assets take on an electronic form that can be transmitted over a network, issues of security will become more pressing.

---

## 2.3

## NETWORK SERVICES

We have just seen that user applications exchange bit streams or messages with widely different traffic characteristics. These applications expect the network to deliver the bit streams or messages within a specific delay and to corrupt only a small fraction of the bits or messages.

Network engineers distinguish two types of information transfer services: connectionless and connection-oriented. We explain that important distinction next.

### 2.3.1 Connection-Oriented Service

When a network implements a connection-oriented service, it delivers messages from the source to the destination in the correct order. Thus, the data transfer in a connection-oriented service appears to take place over a dedicated transmission line, except for the variability in the transmission delay of different packets. A connection-oriented service is required by user applications that expect reliable and ordered transmissions of messages. A CBR or VBR bit stream is delivered by a connection-oriented service.

A connection-oriented service involves three phases: a connection setup phase, a data transfer phase, and a connection teardown phase.

The quality of service (QoS) in some connection-oriented network services specifies whether the transmission is error free and may assign some priority level to packets with the understanding that the network will attempt to transmit high-priority packets before low-priority packets. Thus the delays are likely to be smaller in a high-priority connection-oriented service than in a low-priority connection-oriented service.

Some networks permit a more detailed QoS specification. That specification includes the delay, delay jitter, and packet error rate. It also includes a description of the amount of traffic that the service can transport. These more detailed specifications are required by real-time and interactive applications, as we explained in section 2.2.

When requesting a connection-oriented service in these networks, at the connection setup time the user specifies the QoS that is required by the user application. The network can then determine whether it has sufficient resources available to handle that connection with the requested QoS, and the network can then set aside these resources for the connection. In order to undertake these tasks, the network retains “state” information about existing connections. How these tasks can be carried out is discussed in Chapters 8 and 9.

### 2.3.2 Connectionless Service

When it implements a connectionless service, the network transfers each packet of data to the destination one at a time, independently of the other

packets. Unlike the case with connection-oriented services, the network has no state information to determine whether a packet is part of a stream of other packets. In particular, the network has no knowledge of the amount of traffic that will be sent by the user. Consequently, the network cannot set aside resources that would be needed to achieve a specific quality of service.

Because of this limited information, only a restricted range of service quality can be offered. Typical QoS parameters include a bound on the maximum packet size and service priority: a higher-priority packet is transmitted before a lower-priority packet.

Connectionless service is characterized by the average or typical delay and a specification of the way the service handles errors. Some connectionless services do not indicate when they fail to deliver a message. Other services acknowledge the correct delivery of messages.

Typically, lower layers provide only connectionless service but higher layers may add functionality to provide connection-oriented services.

## 2.4

## HIGH-PERFORMANCE NETWORKS

### 2.4.1

### Traffic Increase

The rate of traffic on the Internet has been increasing rapidly. This increase is caused by new applications, mostly the WWW and e-mail, that have made the network attractive to many users. The volume of traffic that each user generates on the Internet is increasing. E-mail applications let users attach pictures or other large files. Web pages are increasingly rich in animations, audio, and video clips. Inexpensive digital cameras let users exchange photographs over the Internet.

The demand is increasing with the supply of network bandwidth. As the network gets faster, more users take advantage of applications that require that increased speed.

Clearly the trend is to increase the transmission rates of the network links. Low-speed modems (33 Kbps) are giving way to ISDN links (64 Kbps or 128 Kbps) or to digital subscriber loops (384 Kbps or faster) or cable modems (a few Mbps). The long links (tens or hundreds of km) are being upgraded from 1.5 Mbps to 45 Mbps to 155 Mbps to 622 Mbps to 2.4 Gbps and 9.6 Gbps. Multiples of these large rates are made possible by optical multiplexing methods.

To keep up with the increasing link speed, the network switches are also getting faster. This increase of transmission rates is an important component of what we call high-performance networking.

### 2.4.2 High-Performance

We define a high-performance network (HPN) as a communication network that supports a large variety of user applications and that is scalable. In order to support many applications, the network must be able to transfer user traffic at high speed and with low delay. It must be able to allocate resources in ways that match the application requirements. Network organization and management must be flexible so that new applications can be supported as the need arises.

To implement a high-performance network, a number of critical bottlenecks must be addressed. These bottlenecks arise at all layers of the network operations. We examine these bottlenecks in section 2.10 after we explain the key mechanisms that networks implement.

A scalable network can accommodate growing numbers of users without degradation in performance. Growth is usually accommodated by interconnecting distinct networks. The network must be able to provide connectivity over a growing span. The span may be expressed in distance, number of links, or number of subnetworks.

An HPN that supports a wide range of current and future applications and that can accommodate growth is built and managed differently from networks that are designed for a specific application or user population.

At one extreme, phone networks have many nodes and operate over very large distances, but users can transfer data only at low speeds. (Thus phone networks are scalable, but support limited applications.) At the other extreme, a computer backplane bus operates at high speeds but connects only a small number of devices very close to each other. Local area networks or LANs (e.g., Ethernet) can transfer data between tens of nodes at moderate speeds (10 Mbps) over moderate distances (1 km). More recent LANs support speeds of 1 Gbps. Wide area networks or WANs, such as X.25 networks and the Internet, connect hundreds of nodes over hundreds of kilometers, but they operate at limited speeds of a few Mbps or less. Metropolitan area networks (MANs—e.g., DQDB, FDDI, SMDS) run at a higher speed and connect users separated by about 100 km. Frame Relay networks are streamlined versions of X.25 networks that can operate at high speed. The “backbone” telephone network is an HPN: it comprises the switches and links or trunks connecting them, but excludes the low-speed links connecting user telephones to the switches.

The precise values of the user transfer rate, the acceptable delay, the network span, and the number of users that characterize an HPN are somewhat arbitrary. We have in mind a user rate that exceeds 100 Mbps, delays on the order of 100 ms, a span of at least 100 m, and a number of users that can exceed 100. What is essential for a network to qualify as an HPN is for it to be able to support demanding services such as interactive MPEG video and LAN interconnections among many users.

## 2.5 NETWORK ELEMENTS

---

A communication network is a collection of network elements interconnected and managed to support the transfer of information from a user at one network location or node to a user at another node. In this section we discuss the two principal network elements, and we examine how the properties of these elements affect the characteristics of the services that they implement.

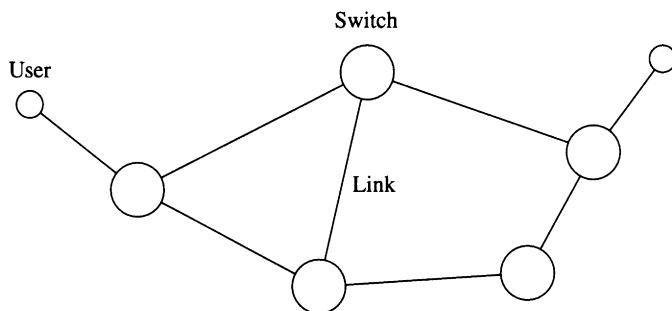
### 2.5.1 Principal Network Elements

The principal network elements are (transmission) links and switches.

A *link* transfers a stream of bits from one end to the other at a certain rate with a given bit error rate and a fixed propagation time. Links are unidirectional. The most important links are optical fiber, copper coaxial cable, and microwave or radio “wireless” links. Optical fiber and copper links are usually point-to-point links, whereas radio links are usually broadcast links. We study links in Chapter 11, focusing on optical links, which are essential for high-speed networks.

Several incoming and outgoing links terminate at a *switch*, which is a device that transfers bits from its incoming links to its outgoing links. Whenever the rate of incoming bits exceeds that of outgoing bits, the excess bits are buffered at the switch. We study switches in Chapter 12, focusing on switches that can handle high-speed traffic.

When we view a network as an interconnection of network elements, we may represent the network as in the graph of Figure 2.1. In the graph, edges denote links, large circles denote switches (including buffers), and small circles denote user nodes where bits are generated or consumed. We will use the names *user*, *source*, *destination*, *station*, and *node* as near synonyms.

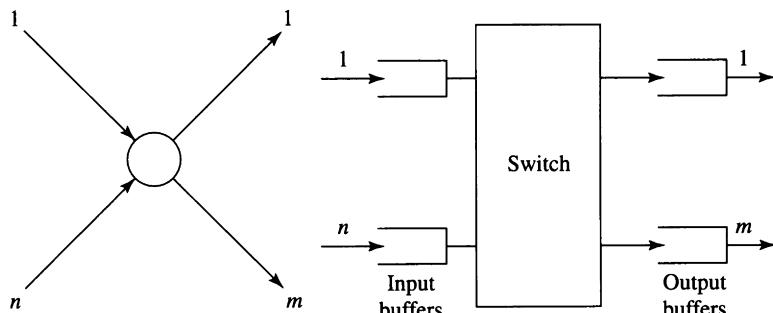


2.1

FIGURE

The principal network elements are transmission links and switches. Transmission links connect user nodes and switches.

Another very important view of an interconnection of network elements is provided by the queuing network model. Consider a switch with  $n$  incoming links and  $m$  outgoing links, as in the left of Figure 2.2. The diagram on the right is the queuing model. Each incoming link's receiver writes into its input buffer, and each outgoing link's transmitter reads from its output buffer. The switch transfers bits or packets from input buffers to the appropriate output buffers. This queuing network model of switches and links is used to describe and evaluate network performance. For example, the queuing delay encountered by a packet in an output buffer is proportional to the number of packets in the buffer in front of it. If packets arrive into a full buffer, there will be packet loss.



2.2

FIGURE

The queuing network model is used for performance analysis. Each switch has a buffer corresponding to each incoming and outgoing link, which is serviced at the link rate.

It is difficult to calculate queuing delay and packet loss. We describe several approaches in Chapter 8.

## 2.5.2 Network Elements and Service Characteristics

A packet generated by a source travels over one link, gets buffered at a switch, is then routed to another link, and so on, until it arrives at its destination.

The delay packets experience through a network depends on the elements that constitute the network, the traffic that goes through these elements, and the way the network is operated.

The detailed analysis of the delay through a network is rather involved. For now, note that we can decompose the total delay into four components:

$$\text{total delay} = \text{TRANS} + \text{PROP} + \text{QD} + \text{PROC}. \quad (2.1)$$

In (2.1) TRANS is the time required to transmit a packet, so

$$\text{TRANS} = (\text{packet size}) / (\text{transmission speed}). \quad (2.2)$$

For example, for a 10,000-bit packet and a transmission speed of 1 Mbps, TRANS is 10 ms. PROP is the signal propagation time, so

$$\text{PROP} = \frac{(\text{distance from source to destination})}{(\text{speed of electrical or optical signal})}. \quad (2.3)$$

Propagation time for an electrical or optical signal is between 3.3 and 5 microseconds per kilometer ( $\mu\text{s}/\text{km}$ ). QD is the queuing delay in the switch. It occurs whenever the bit rate of the traffic coming into the switch exceeds the capacity of the outgoing link. The excess bits are queued in the switch buffers. In contrast with the other two sources of delay, queuing delay is significantly affected by the network control policy. Finally, PROC is the processing time required by the network switches. We will assume that this processing time is negligible.

Suppose as a rough rule of thumb that the network is controlled so that each packet coming into the switch has to wait on average for four previous packets to be transmitted. Then the average queuing delay is four times the transmission delay, so

$$\text{total delay} = 5 \times \text{TRANS} + \text{PROP}. \quad (2.4)$$

By combining the expressions (2.2) through (2.4), we see how the delay depends on the transmission rate and length of the links and on the length or size of messages.

### 2.5.3 Examples

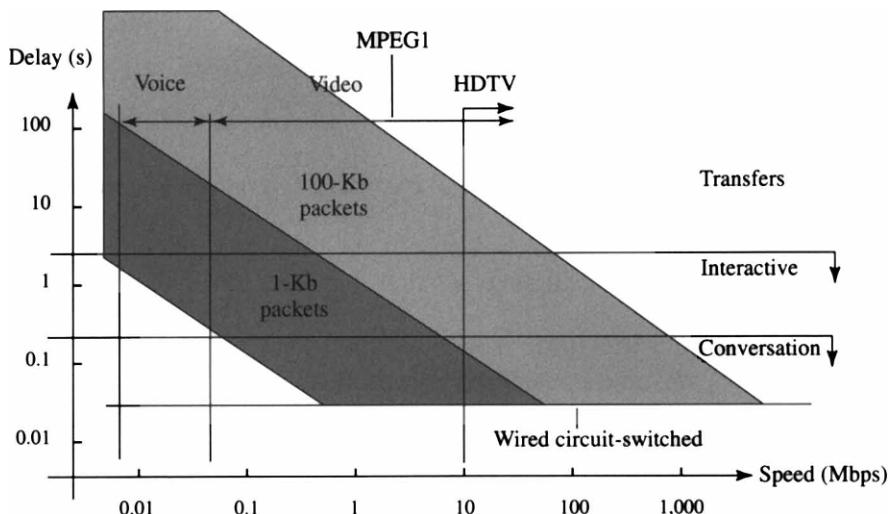
We illustrate how we can use the preceding analysis in three examples.

Consider a potential telecommuting application in which a company employee responds to customer billing inquiries from a terminal at home. When a customer calls, an automatic call-transfer service transfers the call to the employee's home. The employee then requests the customer's record from the company's database computer. The record is transferred by the network and displayed on the terminal. The employee then answers the customer's questions and updates the record as necessary. For the response to be satisfactory to the customer, the retrieval of the record and its display should not take more than one second, say. A screen full of text takes about 16,000 bits, so the network response will be satisfactory provided the transmission speed of  $B$  bps is such that

$$\text{total delay} = 5 \times 16,000 \times B^{-1} + \text{PROP} < 1.$$

In this example, we may neglect PROP, which is a few  $\mu\text{s}$ . We see then that this application needs a transmission speed of 80 Kbps. This rate is supported by a DSL service that costs about \$50 per month (1999 prices). Comparing this cost with the savings to the employee and employer resulting from fewer work commute trips, less office and parking space, and more flexible work schedules, we can imagine a large potential demand for communication services needed to support telecommuting. (It is estimated that a telecommuter working at home 1 to 2 days per week can save most companies \$6,000 to \$12,000 a year because of increased productivity, lower staff turnover, and reduced office space. The annual cost per telecommuter is estimated to be up to \$10,000 for equipment, space at home, and phone services.)

As another telecommuting example, consider a design engineer or architect using a workstation at home. Most of the material needed for work is stored in the workstation's local disk. However, from time to time, the engineer needs to retrieve or replace a large file (e.g., several bit maps) stored in a file server at the workplace. Such a file is about 3 MB long, and if the workstation is connected by a 1-Mbps link, the retrieval would take about 24 seconds. This may be adequate if such transmissions are infrequent. However, if 10 transmissions



2.3

FIGURE

Delay as a function of transmission rate for a U.S.-wide network with 10 nodes and a queue at each node of 1 to 100 packets. Delay for the circuit-switched connection equals the propagation time.

every minute are needed, this may require a 4-Mbps link, which at current costs may make this telecommuting application uneconomical. (At the workplace, the engineer's workstation and file server are connected by a relatively inexpensive 10-Mbps local area network such as Ethernet.)

The third example provides a more general illustration of the requirements that applications place on the delay and speed of bearer services. Figure 2.3 compares three bearer services that transport data across the United States through 10 switches at various combinations of speed and delay.

The first service is implemented by a circuit-switched network using wired (optical or copper coaxial) links. The end-to-end delay equals the propagation delay, PROP, independent of the speed. This delay is about 25 ms (5,000 km  $\times$  5  $\mu\text{s}/\text{km}$ ). This service is summarized by the horizontal line labeled "Wired circuit-switched."

The second and third services are implemented by a packet-switched network using 1-kilobit (Kb) and 100-Kb sized packets, respectively. If at each switch, a packet encounters between 0 and 100 packets waiting in queue, then the queuing delay (through the 10 switches) is between 0 and 1,000 packet transmission times. The transmission time, TRANS, is 1,000/bit rate for the

smaller and  $10^5$ /bit rate for the larger packet. So for the second service the total delay is

$$\text{PROP} < \text{total delay} < \text{PROP} + 10^6/\text{bit rate},$$

which is the lower shaded region in the figure. For the third service the total delay  $D$  is

$$\text{PROP} < \text{total delay} < \text{PROP} + 10^8/\text{bit rate},$$

which is the upper shaded region.

We now consider three applications: a conversation, an interactive exchange, and a transfer of a voice or video file. The conversation can tolerate a delay of 250 ms, the interactive exchange can accept a few seconds of delay, and the file transfer can tolerate a large delay. These bounds are shown in the figure. Finally, depending on the compression technique used, voice will generate traffic at a rate between 8 and 64 Kbps, video between 64 Kbps and 10 Mbps, and HDTV will generate higher-speed traffic.

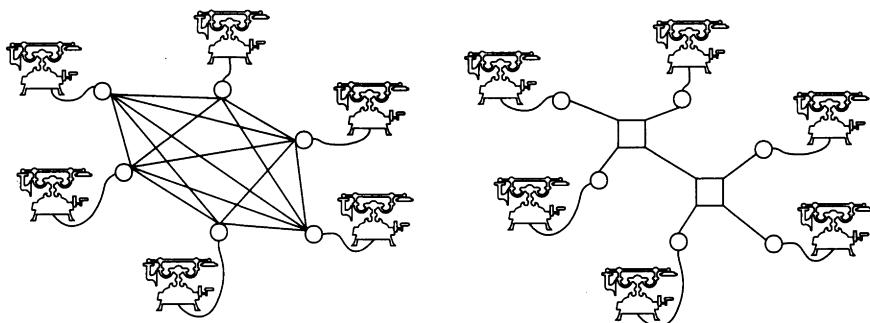
When we compare the delay and speed requirements of the application with the characteristics of the three services, we see that at speeds above 10 Mbps there is little difference between circuit-switched connections and a 1-Kb packet-switched service. With 100-Kb packets, higher speed is needed. For example, at T3 speed of 45 Mbps the delay of a 100-Kb packet service is indistinguishable from a circuit-switched connection, provided queues are shorter than 100 packets. For interactive and transfer applications, the advantage of a circuit-switched connection over the packet-switched services similarly disappears at higher speeds.

## 2.6

## BASIC NETWORK MECHANISMS

A network's bearer services comprise the end-to-end transport of bit streams, in specific formats, over a set of routes. These services are differentiated by quality: speed, delay, errors. They are produced using five basic mechanisms: multiplexing, switching, error control, flow control, congestion control, and resource allocation. We discuss those mechanisms in this section.

As we noted in section 1.2.2, there are economies of scale in transmission. That is, the cost of a transmission link connecting two nodes does not increase in proportion to its capacity. Individual users located at those two points, however, typically need a small transmission bandwidth for short durations.



2.4

FIGURE

The left-hand part of the figure shows a fully connected network. The right-hand part of the figure shows a network where some links are shared. The sharing of links is made possible by multiplexing and switching.

*Multiplexing* combines data streams of many such users into one large bandwidth stream for long durations. Users thereby can share in the scale economies of transmission. However, users are not concentrated in a few locations; they are geographically dispersed. *Switching* allows us to bring together the data streams of these dispersed users. (In essence, the communication network industry makes profits by installing large transmission capacity and by “renting” this capacity to users in smaller amounts using multiplexing and switching.)

In the left side of Figure 2.4 we see a network with no multiplexing or switching. Each pair of phones is connected by a dedicated link of capacity 64 Kbps needed to carry one voice conversation. In this network, the average number of links per phone, and hence the average cost, increases with the number of nodes. Since a telephone is engaged in at most one conversation at any time, link utilization, that is, the fraction of time the link is busy, decreases as the number of nodes increases. Thus the cost per phone grows with the size of the network. This network wastes resources.

The network on the right employs multiplexing and switching. Each phone is connected by a dedicated access link to one of two local switches, which are connected by a single link called a trunk. (In telephone networks, a switch is located in a *central office*; a link between two switches is called a *trunk*; a link between a subscriber telephone and a switch is called an *access line* or *subscriber loop*.) The access line capacity is 64 Kbps, and the trunk capacity is a multiple of 64 Kbps. (In this example it may be  $2 \times 64$  Kbps.)

Depending on its capacity, a trunk can carry several voice conversations simultaneously by multiplexing. A conversation between two phones will occupy only two access lines—if both phones are connected to the same local switch—or, in addition, it will occupy a fraction of the trunk capacity. It is the

task of the switch to determine whether a call originating from a local phone is destined for another local phone or for a phone connected to the remote switch. In this network, there is one access line per phone, but the switch and trunk capacities grow much less rapidly than the number of phones. As a result, the average cost of the network per user decreases with the number of phones. This decreasing cost structure of communication networks is made possible by multiplexing and switching.

Another important mechanism is *error control*. All transmission links occasionally corrupt the messages they transmit. Although carefully designed and maintained links have a very small bit error rate (e.g.,  $10^{-12}$ ), even the rare errors in the transmissions may not be acceptable. Moreover, in addition to transmission errors, a message may fail to reach its destination because it arrived at a switch whose buffer was full and so the message was discarded. It is therefore important for the network to control such errors. We explain two methods that networks use to control errors later in this section.

A source must prevent overwhelming the receiver by sending messages faster than the receiver can store or process them. *Flow control* is a mechanism that enables the receiver to pace the transmissions of the source.

The end-to-end delay is the sum of a fixed propagation delay and a variable queuing delay. In order to keep this delay within an acceptable range, the rate at which packets enter the network must be controlled. *Congestion control* is the generic name for a set of mechanisms designed to limit the rate or number of packets introduced into the network by a source or a switch. If the congestion control mechanism does not function properly, an excessive number of packets may accumulate in the switch buffers causing unacceptable delay or loss.

We have seen that some applications require the network to provide a minimum bandwidth to ensure acceptable performance. For example, a voice conversation requires 64 Kbps and MPEG1 requires 1.5 Mbps. A variable bit rate application will require a guaranteed combination of minimum bandwidth and buffers. Because network resources—link bandwidth and switch buffers—are shared by many applications at the same time, *resource allocation* mechanisms must be designed to ensure that each application receives the necessary resources to maintain its quality of service.

Multiplexing is carried out in switches or specialized hardware. Switching and resource allocation are implemented in switches. Error control and flow control are implemented in switches and host computers.

## 2.6.1

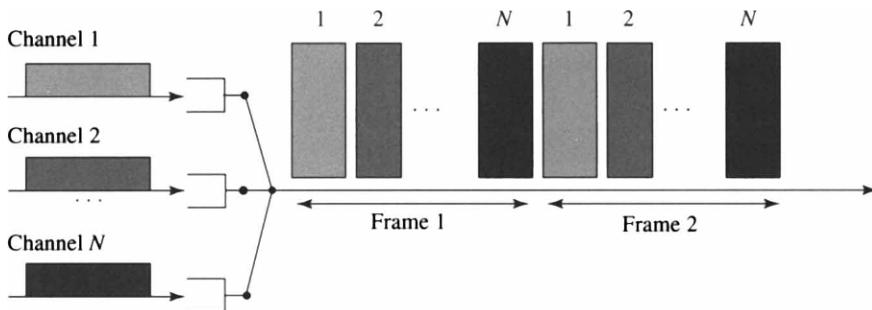
### Multiplexing

We now explain three important techniques for multiplexing  $N$  incoming channels onto one outgoing channel. These are time-division multiplexing,

statistical multiplexing, and frequency-division multiplexing. For our present purposes a *channel* is a communication link of fixed capacity measured, say, in bits per second (bps). That is, each second of time on the channel is divided into a number of intervals called *bit times*, the number being equal to the channel capacity. For example, in a 1-Mbps channel, a bit time is  $1 \mu\text{s}$  long. Each bit time may be occupied by an information or data bit, or it may be empty. We also say, accordingly, that the channel is *busy* or *idle* at that time. The average data rate divided by channel capacity, or the fraction of time the channel is busy, is the channel *utilization*. Utilization is equal to 100% only if the instantaneous data rate is always equal to the channel capacity.

*Multiplexing* is the process by which information bits from  $N$  incoming channels are transferred into bit times on one outgoing channel. *Demultiplexing* is the reverse process: the information bits on one incoming multiplexed channel are separated and transferred onto  $N$  outgoing channels. The multiplexed outgoing channel contains some extra bits (in addition to the incoming channels' data bits) that the demultiplexer uses to determine which data bits belong to which incoming channel.

*Time-division multiplexing* or TDM is illustrated in Figure 2.5. In TDM, the capacity of the outgoing channel is divided into  $N$  logical channels, and data in each of  $N$  incoming channels is placed in a designated outgoing logical channel. This is achieved as follows. Time on the outgoing channel is divided into fixed-length intervals called *frames*. Frames are delimited by a special bit sequence called a *framing pattern*, not shown in the figure. Time in each frame is further subdivided into  $N$  fixed-length intervals called *slots* (this name is not always used). Thus each frame consists of a sequence of slots: slot 1, slot 2, . . . , slot  $N$ .



2.5

FIGURE

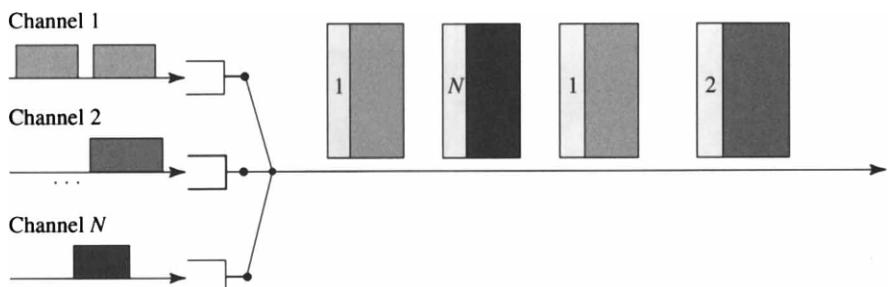
When a communication link is shared by time-division multiplexing, time is divided into frames. Each frame is divided into time slots that are allocated in a fixed order to the different incoming channels.

(A slot is usually 1 bit or 1 byte wide.) A logical channel occupies every  $N$ th slot. There are thus  $N$  logical channels. The first logical channel occupies slots 1,  $N + 1$ ,  $2N + 1$ , . . . ; the second occupies slots 2,  $N + 2$ ,  $2N + 2$ , . . . ; and so on.

The multiplexer operates as follows. The data bits in each incoming channel are read into a separate FIFO (first in, first out) buffer. The multiplexer reads this buffer in sequence for an amount of time equal to the corresponding slot time: buffer 1 is read into slot 1, buffer 2 is read into slot 2. (If there are not enough bits in a buffer, the corresponding slot remains partially empty.) The bit stream of the outgoing channel is easily demultiplexed: the demultiplexer detects the framing pattern from which it determines the beginning of each frame, and then each slot.

TDM is easy to implement. The overhead due to extra framing bits is small. However, the utilization of the outgoing channel may vary a great deal depending on the burstiness of the incoming data streams. To see this, observe that the capacity of the outgoing channel must be as large as the sum of the capacities of the  $N$  incoming channels. As a result, the utilization of the outgoing channel will be low or high accordingly as the utilization of the incoming channels is low or high. TDM leads to high utilization if the incoming data is not bursty. Thus TDM is ideal for constant bit rate traffic.

*Statistical multiplexing* or SM, illustrated in Figure 2.6, is most effective in the case of bursty input data. As in TDM, the data bits in each incoming channel are read into separate FIFOs. The multiplexer reads each buffer in turn until the buffer empties. (It is customary to call the data read in one turn a data *packet*.) In TDM each FIFO is read for a fixed amount of time—one slot—and

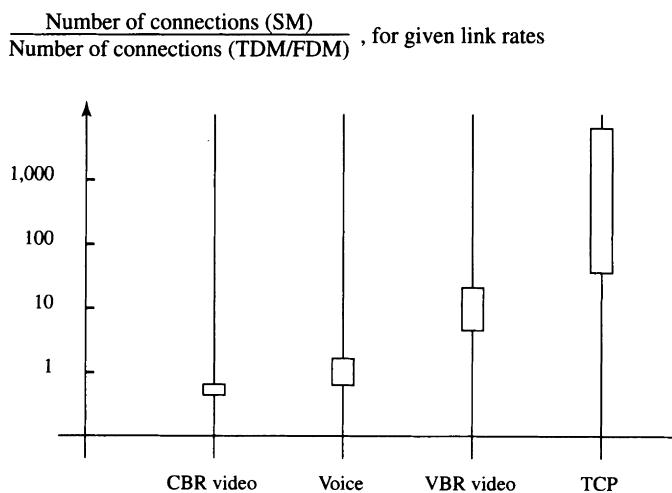


2.6  
FIGURE

In statistical multiplexing, the multiplexer visits the incoming channel buffers in some order. The multiplexer empties a buffer before moving to the next one. The buffer contents are tagged to indicate their incoming channel. An idle channel does not waste transmission time.

so each incoming channel is allocated a fixed fraction of the outgoing channel capacity, independent of the data rate on that channel. By contrast, in SM, the capacity allocated to each incoming channel varies with time, depending on its instantaneous data rate: the higher the rate, the larger the capacity allocated to it at that time. As a result, the capacity of the outgoing channel needs to be only as large as the sum of the average data rates of the incoming channel, which, for bursty traffic, may be much smaller than the sum of the peak data rates. Hence the capacity of the outgoing channel may be smaller than the sum of the incoming channel capacities. We call the ratio of the total incoming capacity to the total outgoing capacity the *multiplexing gain*. This gain is unity for TDM but can be much larger for SM. Figure 2.7 illustrates the possible multiplexing gains for SM relative to TDM and FDM for various applications.

As we have seen, in SM the size of packets read from each FIFO can vary across channels and over time within each channel. Therefore, the demultiplexer cannot sort the packets belonging to different channels merely from their position within a frame. Consequently, additional bits, which delimit each packet and identify the corresponding incoming channel or source, must be added to each packet. The resulting overhead is significantly larger than under TDM. It also becomes more difficult to implement the multiplexer (which must now add the packet delimiter and channel or source identifier) and the



2.7  
**FIGURE**

Statistical multiplexing can achieve much higher multiplexing gain relative to TDM and FDM, especially for bursty traffic.

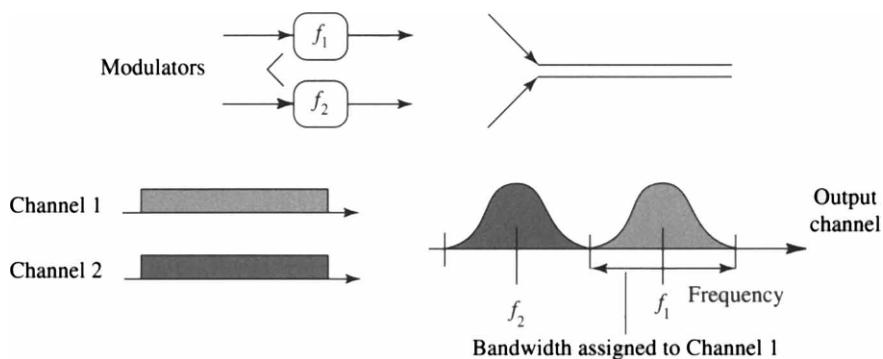
demultiplexer (which must locate and decode those bit patterns). These increases in complexity and overhead must be balanced against high utilization in the face of bursty data to determine whether SM or TDM is more efficient. In general, constant bit rate traffic, such as voice, fixed-rate video, and control and sensor signals, are better handled by TDM, whereas message traffic such as database transactions and variable bit rate video are better handled by SM. Not surprisingly, telephone networks use TDM, whereas computer communication networks use SM.

In some cases, one may want to treat different packet streams differently. That is, instead of transmitting the packets in their order of arrival at the statistical multiplexer, some other strategy may be preferable. A number of scheduling strategies have been designed. A priority strategy always transmits from the first nonempty queue in a fixed order. A weighted round-robin strategy multiplexes  $k$  queues by first serving  $M(1)$  packets from queue 1, then  $M(2)$  from queue 2, and so on until it serves  $M(k)$  packets from queue  $k$  and then repeats the cycle. If any queue  $i$  runs out of packets before  $M(i)$  packets are transmitted from the queue, then the transmitter continues with the next queue. The objective of these strategies and their variations is to provide a differentiated service to different classes of applications.

*Frequency-division multiplexing* or FDM is illustrated in Figure 2.8. The frequency band of the outgoing channel is divided into distinct fixed bands, one for each incoming channel. The signal in each incoming channel is modulated to fit into its assigned band. The signal on the outgoing channel is simply the sum of these modulated signals. Thus the bandwidth of the outgoing channel must be greater than or equal to the sum of the bandwidths of the incoming channel. (In this sense, FDM is similar to TDM.) For demultiplexing, the FDM signal is passed through an appropriate filter. By demodulating the filter output, we obtain the appropriate input signal. FDM is used in AM and FM radio and TV broadcast as well as in CATV distribution. It is also used in cellular radio. FDM is more flexible than the other two multiplexing schemes. Indeed, one incoming channel may contain an analog signal, and another may be digital.

FDM has two disadvantages. First, it is wasteful of bandwidth since the frequency bands assigned to an incoming channel must be separated by a "guard band" from the other channels. Second, if the transmission link exhibits significant nonlinearities, there will be "cross-talk" among the different signals, leading to errors.

We have used bandwidth (or size of a link's frequency band in Hz) and speed or bit rate of a link (in bps) interchangeably, but that is not strictly correct. A (digital) bit stream is converted into an analog signal by a modulation scheme like phase-shift keying (PSK). The frequency spectrum of the result-



2.8

FIGURE

In frequency-division multiplexing, the frequency band is divided into distinct fixed bands, one for each incoming channel. The signal in each incoming channel is modulated to fit into its assigned band.

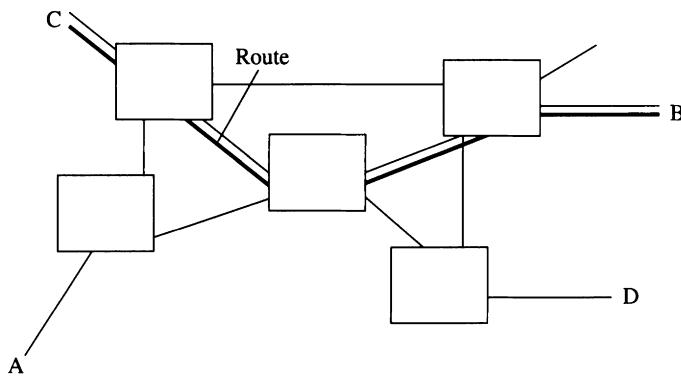
ing analog signal must lie within the link's frequency band. As a result the bit rate is proportional to the bandwidth. The ratio bit rate/bandwidth measured in bps/Hz is called the *spectral efficiency*, which typically ranges between 0.5 and 8.0. The maximum value of the achievable spectral efficiency is given by Shannon's theorem as a function of the noise in the communication channel. When bandwidth is scarce, as in wireless radio and satellite channels, communication engineers spend a lot of effort designing modulation schemes to increase spectral efficiency.

*Code-division multiplexing* is yet another method for multiplexing different bit streams on a common link. Some cellular networks use this multiplexing method. We describe that modulation method in the chapter on wireless systems (Chapter 7).

## 2.6.2

## Switching

We now describe the two most important switching techniques: circuit switching and packet switching. When a network is to transfer a stream of data from a source to a destination (e.g., from *A* to *D* or *C* to *B* in the following figures), it must assign to the stream a *route*, that is, a sequence of links or channels connecting the source to the destination, and then allocate to the stream a portion of the capacity or bandwidth in each channel along the route to be used to transfer the stream. Those decisions are implemented in switches. (Route selection and bandwidth allocation are examples of resource allocation.) The name *switch* is used in telephony; in computer communications, the device



**2.9**  
**FIGURE**

In order to transmit information, a circuit-switched network finds a route along which it has free circuits. The network connects the circuits together and reserves them for the transmission.

that performs routing is also called a *router*. We shall use *switch* and *router* interchangeably.

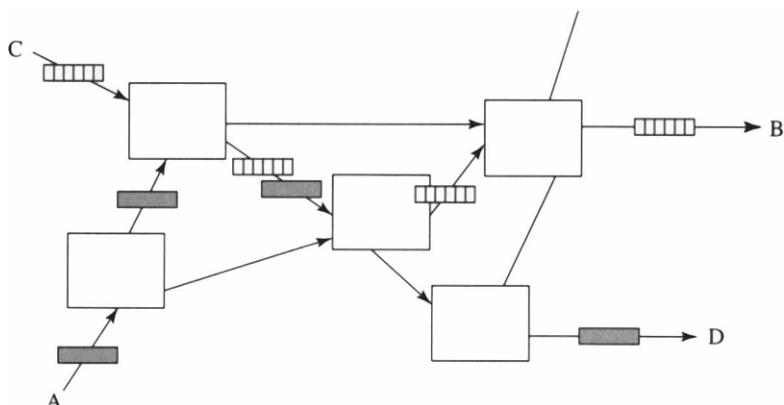
In *circuit switching*, illustrated in Figure 2.9, the route and bandwidth allocated to the stream remain constant over the lifetime of the stream. Further, the capacity of each channel is divided into a number of fixed-rate logical channels, called *circuits*. The division is usually accomplished by TDM. Thus circuit switching involves assigning to the bit stream a route and one circuit in each link along the route. This assignment is made before any bits are transferred in a phase known as *call* or *connection setup*. At the end of this phase, data transfer is carried out. When the transfer is complete, the route and the circuits are deallocated. That phase is called *connection teardown*.

Circuit switching thus involves three phases: (1) the source makes a *connection* or *call request* to the network, the network assigns a route and one idle circuit from each link along the route, and the call is then said to be admitted (if the network is unable to make this assignment, the call is rejected); (2) data transfer now occurs—the duration of the transfer is called the *call holding time*; (3) the call is then torn down. The switch computers maintain information about which circuits are busy (i.e., currently allocated to calls in progress) and the routing tables. The computers also execute algorithms implementing call admission policies and routing strategies. They also record call duration and other statistics needed for purposes of administration, billing, and maintenance.

Circuit switching is easy to implement relative to other schemes, but because a stream is assigned a fixed-rate circuit, capacity utilization may be low if the data stream is bursty. It is therefore used in voice networks but not in networks designed for data transfer. Since circuit switching assigns a fixed bandwidth to a call, there is no queuing delay, and we can guarantee ahead of time the maximum end-to-end delay from source to destination experienced by the data stream. This guaranteed-delay feature may be essential in control, videoconferencing, and other real-time applications. From (2.1), this delay equals the transmission plus propagation times.

Figures 2.10 through 2.12 illustrate *packet switching*. The data stream originating at the source is divided into packets of fixed or variable size. The time interval between consecutive packets may vary, depending on the burstiness of the stream. As the bits in a packet arrive at a switch or router, they are read into a buffer. When the entire packet is stored, the switch routes the packet over one of its outgoing links. The packet remains queued in its buffer until the outgoing link becomes idle. This *store-and-forward* technique thereby introduces a random queuing delay at each link; the delay depends on the other traffic sharing the same link. (Packets from different sources sharing the same link are statistically multiplexed.) The total delay at a switch is the queuing delay plus the time taken to transmit the entire packet.

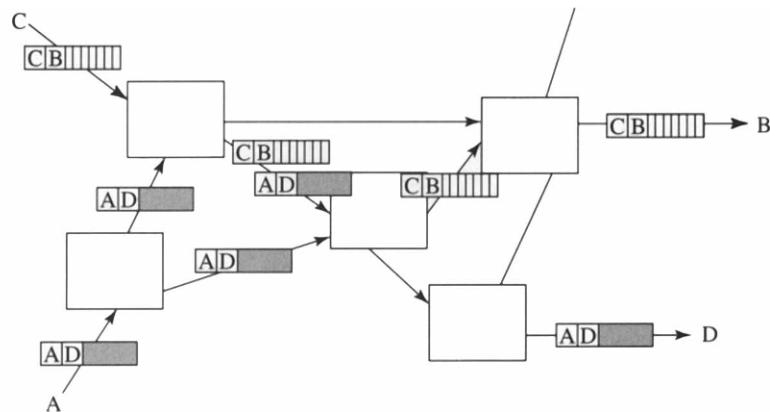
The routing decision is determined in one of two ways: datagram and virtual circuit. In *datagram* packet networks, each packet within a stream is



2.10

FIGURE

A packet-switched network first divides the information it has to transmit into packets. The packets are then sent along links that are multiplexed statistically.

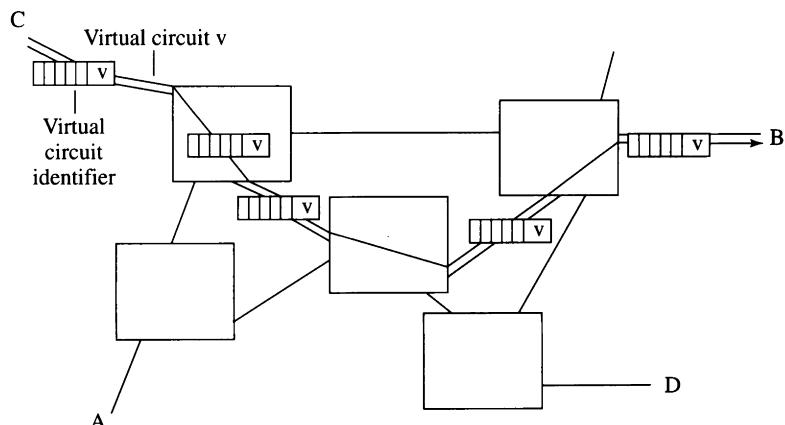


2.11

FIGURE

A datagram network transports packets individually. Each packet contains its source and destination addresses.

independently routed, as illustrated in Figure 2.11. A routing table stored in the router (switch) specifies the outgoing link for each destination. The table may be static, or it may be periodically updated. In the latter case, the outgoing link assigned to a destination depends on the router's estimate of the shortest



2.12

FIGURE

A virtual circuit network transports all the packets of the same connection along the same path, called a virtual circuit. Capacity along the virtual circuit is not reserved for a connection.

path to the destination. Since the estimate may change with time, consecutive packets may be routed over different links. (Observe in the figure that packets from A to D travel over two routes.) Therefore each packet must contain bits denoting the address of the source and destination. This may be a significant overhead (since the addresses are often quite long) if the average packet size is small; the overhead may be negligible if the packet size is long, but then the packet transmission time and the queuing delay are also long.

In *virtual circuit* packet networks, a fixed route is selected before any data is transmitted in a call setup phase similar to circuit-switched networks. (See Figure 2.12.) However, there is no notion of a fixed-rate circuit or logical channel. All packets belonging to the same data stream follow this fixed route, called a virtual circuit. Packets must now contain a virtual circuit identifier; this bit string is usually shorter than the source and destination address identifiers needed for datagrams. However, the call setup phase takes time and creates a delay not present in datagram packet networks.

Datagram switching achieves higher link utilization than circuit switching, especially when traffic is bursty. It is used in data networks. Datagram switching, however, also has potential disadvantages in comparison with circuit switching. First, the end-to-end delay may be so large or so random as to preclude applications that demand guaranteed delay. Second, the overhead due to source and destination identifiers and bits needed to delimit packets may waste a significant fraction of the transmission capacity if the packets are very short. Since consecutive packets may travel over different routes, packets may not arrive at the destination in the same sequence in which they were sent. The host at the destination may then have to buffer several packets and resequence them, adding to the end-to-end delay. Lastly, a datagram switch does not have the state information to recognize if a packet belongs to a particular application. Hence the switch cannot allocate resources (bandwidth and buffers) that the application may require. Thus it is difficult to implement sophisticated resource allocation schemes in a datagram network.

Virtual circuit switching is a compromise between datagram and circuit switching. The overhead is comparable to circuit switching. Since packets arrive at the destination in order of transmission, no resequencing is needed. Statistical multiplexing of packets at the router or switch can achieve better utilization than in circuit switching. But the utilization may be lower than under datagram switching, since routing decisions in the latter may be changed at the time scale of individual packet durations. Under virtual circuit switching those decisions can be changed only at the time scale of call durations. Lastly, since packets contain their virtual circuit identifiers or VCIs, the switch can allocate resources depending on the VCI. During the connection setup phase

the switches may be notified that a particular VCI should be given extra resources.

In circuit switching, bit streams are allocated fixed-capacity circuits, and incoming bit streams are time-division multiplexed into outgoing streams. Only a tiny amount of buffering is needed at the circuit switch—enough to compensate for short-term fluctuations in the frequency and phase of the incoming and outgoing bit streams. In packet switching, the rate of the incoming stream may exceed the capacity of the outgoing link for significant time intervals. Thus packet switches must be able to buffer the excess incoming traffic. Consequently, buffer management is another task performed at the switch in addition to routing and multiplexing.

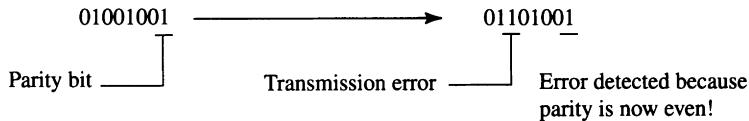
### 2.6.3 Error Control

Figure 2.13 illustrates the basic ideas behind error control. To control errors, a transmission link can use two methods: *error detection* and *error correction*.

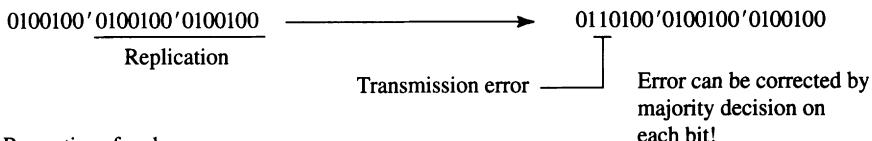
A simple error-detection method is the parity bit used, for example, in the RS-232-C serial line discussed in section 1.1.2. When the link uses that method, the transmitter appends a 1 or a 0 to the packet it transmits so that the resulting string (packet + parity bit) contains a number of 1s that has an agreed parity, say odd. For instance, if the original packet is 0100100, then the transmitter appends a 1 and transmits the resulting string 01001001, which contains an odd number (three) of 1s. If transmission errors modify the string into one whose number of 1s is even, then the receiver detects an error. For instance, if the receiver gets the string 01101001, then it knows that the transmission modified at least one bit. This parity bit method does not detect errors that modify an even number of bits. Thus, this method is not very reliable for long packets: it is not unlikely that transmission errors modify 2 or 4 bits in long packets. Communication engineers developed error-detection methods that detect all but very unlikely errors by adding more than one error-detection bit to the packets. Two such methods are the *cyclic redundancy check* (CRC) and the *checksum code* (CKS). We discuss the CRC method in the next two paragraphs. The discussion is highly technical, and the uninterested reader may skip it without loss of continuity.

The CRC method works as follows. A binary word  $G$  forms the basis of the CRC calculation. Standards specify the word  $G$  that should be used for different networks. Denote by  $r + 1$  the number of bits of  $G$ . Assume that we want to transmit a packet that consists of the finite string of bits  $P$ . We denote by  $P2^r$  the binary word obtained by appending  $r$  0s to the right of the packet  $P$ . By performing a long division, an electronic circuit in the transmitter calculates

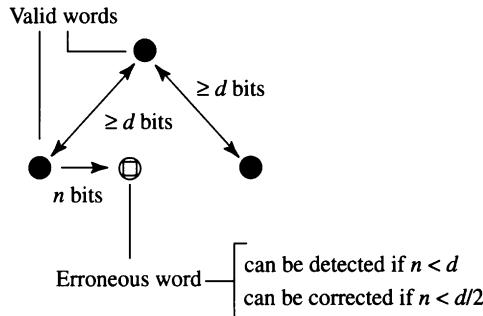
Error detection with odd parity:



Error correction with replication:



Properties of codes



2.13

FIGURE

The figure illustrates error detection and error correction. The bottom panel shows that the error-control properties of the codes depend on the minimum distance between valid codewords.

the remainder  $R$  of the division of  $P2^r$  by  $G$ . This long division is calculated by performing the additions of the binary words bit by bit modulo 2 and without carry. With these rules,  $10011 + 01011 = 11000$ . By definition, the remainder  $R$  has  $r$  bits and is such that  $P2^r = A \cdot G + R$ . The binary word  $R$  is the CRC code that the transmitter appends to the packet. That is, the transmitter sends the binary word  $P2^r + R$ . Note that, because of the rules used to perform the operations,  $P2^r + R = A \cdot G + R + R = A \cdot G$ . Thus, every transmitted packet, together with its CRC bits, is an exact multiple of  $G$ . If the receiver gets a packet that is not a multiple of  $G$ , then it knows that transmission errors corrupted the packet.

We can represent transmission errors by a binary word  $E$  whose 1s indicate which bits the transmission corrupts. With this representation the packet that the receiver gets is  $A \cdot G + E$ . This received packet is a multiple of  $G$ , and the receiver does not detect the errors if and only if  $E$  is a multiple of  $G$ . Thus, to

be a robust error-detection code, the CRC should use a binary word  $G$  that is not likely to divide an error word  $E$ . For instance, we can show that if  $G = 1'0001'0000'0010'0001$ , then  $E$  cannot be a multiple of  $G$  if  $E$  has fewer than 32,768 bits and has fewer than four 1s. Thus, the 16-bit CRC that uses this specific  $G$  detects all errors that corrupt up to 3 bits in a packet of up to 32,768 bits.

The middle panel of Figure 2.13 illustrates a simple error-correction method that transmits every packet three times. By performing a majority vote on every bit, the receiver can correct transmission errors as long as they affect each packet bit in at most one of the three transmissions. Thus, if the third bit in the packet is a 0 and if one of the transmissions modifies it into a 1, then the receiver can decide that the transmitter sent a 0 and thereby correct the error.

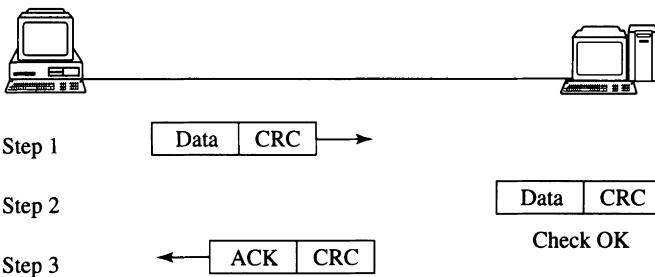
This replication method is wasteful and is not very robust. Communication engineers have designed better methods, including the *Bose-Chaudhury-Hocquenghem* (BCH) and the *Reed-Solomon* (RS) codes.

In general terms, an error-control code adds  $R$  bits to a packet of  $M$  bits. The code computes the  $R$  bits from the original  $M$  bits so that the transmitter can send only  $2^M$  different strings of  $M + R$  bits, called *codewords*. Assume that any two codewords differ by at least  $d$  bits. (See bottom panel of Figure 2.13.) Then, the receiver can detect transmission errors that modify fewer than  $d$  bits. Indeed, such errors cannot modify a codeword into another codeword, and the receiver detects the error when it finds that the received string is not a codeword. The receiver can correct errors that modify fewer than  $d/2$  bits because the transmitted codeword is the codeword that differs the least from the received string.

A potentially useful form of error correction for packet networks is to use a code that enables the receiver to recover correct information provided that it gets any  $n$  out of  $n + m$  packets. That is, instead of adding redundant information to a packet to protect it, one adds redundant packets to a group of packets to protect them.

Computer networks today use error detection. The transmitter retransmits packets that do not arrive intact at the receiver. The transmitter and receiver follow a specific set of rules, a *protocol*, for making sure that the receiver eventually gets every packet correctly and that it discards corrupted packets and duplicated correct copies. These protocols use timers and acknowledgments.

Figure 2.14 sketches a typical implementation of such a protocol. Before sending a packet, the transmitter makes a copy that it keeps. The transmitter then sets a count-down timer to a specific value and sends the packet. We say that the packet *times out* when the count-down timer reaches the value 0. If the receiver gets the intact packet, then it sends back an acknowledgment to the

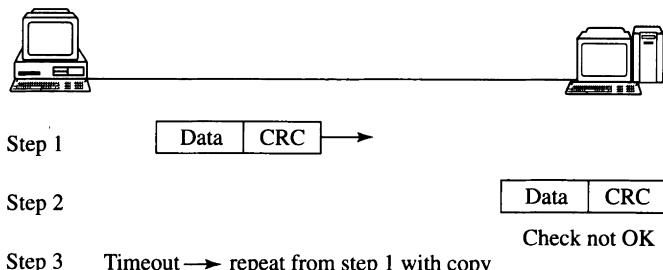


**2.14**  
**FIGURE**

The figure shows an error-free transmission. The transmitter sends the packet (step 1). When it receives the packet, the receiver verifies that it is correct (step 2). The receiver sends an acknowledgment of the packet to the transmitter (step 3).

transmitter. If the transmitter gets the acknowledgment of the packet before the packet times out, then the transmitter knows that the receiver got the packet. In that case, the transmitter discards its copy of the packet and repeats the procedure with the next packet.

If the receiver gets a corrupted packet (see Figure 2.15), then it does not send an acknowledgment. When the packet times out, the transmitter assumes that something went wrong during the packet transmission, and it repeats the procedure with a copy of the packet. The packet also times out when its acknowledgment is corrupted during its transmission.



**2.15**  
**FIGURE**

The figure shows a transmission corrupted by errors. The transmitter sends the packet (step 1). When it receives the packet, the receiver finds out that it is incorrect by checking the error-detection bits in the CRC (step 2). The receiver does not send an acknowledgment of the packet to the sender. When its packet times out, the sender transmits a copy of the packet (step 3).

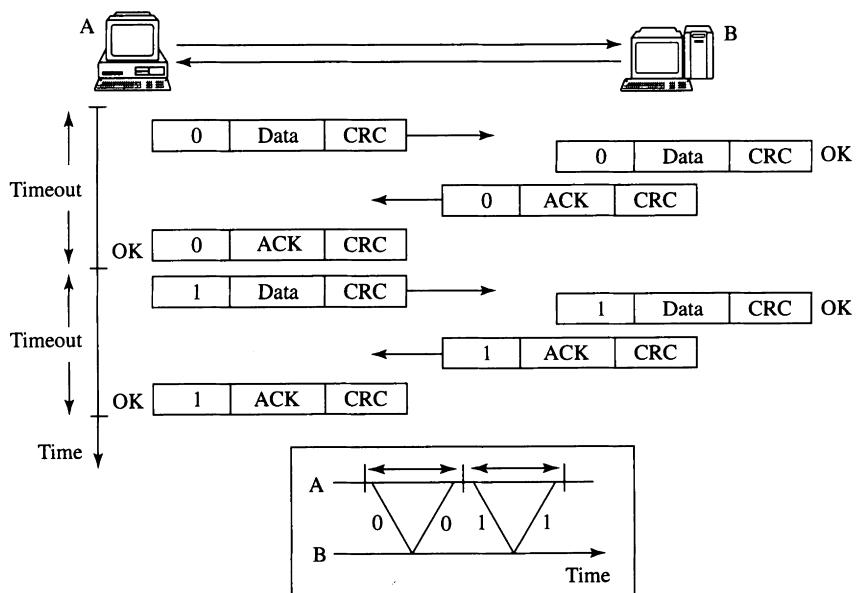
If there is no permanent hardware problem and if the initial value of the timer is large enough, the transmitter eventually gets an acknowledgment before it times out and every packet eventually reaches the receiver.

We explain the retransmission protocols that networks use below. For real-time applications and storage, the network should use error correction instead of error detection.

### *Alternating Bit Protocol*

The simplest retransmission protocol is the *Alternating Bit Protocol* (ABP). The sender numbers the packets alternately 0 and 1. The receiver acknowledges every correct packet that it receives with the same number as the received packet. The sender waits for the receiver's acknowledgment. Figure 2.16 shows the sequence of events when there is no transmission error.

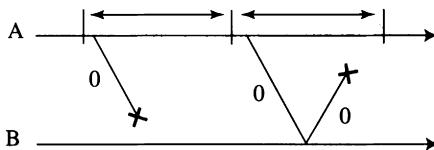
If the acknowledgment does not reach the sender before the packet times out, then the sender retransmits the packet (with the same number), as illus-



2.16

FIGURE

The figure shows packet 0 arriving correctly at the receiver, which then sends an acknowledgment 0. The acknowledgment arrives before the timeout value. The transmissions then repeat with sequence number 1. This sequence of transmissions is summarized in the timing diagram at the bottom of the figure.



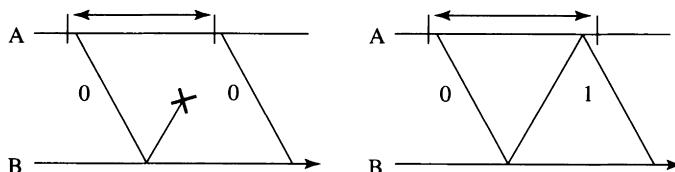
2.17

FIGURE

A cross on the downward line marked 0 represents a transmission error that corrupts packet 0. The receiver does not acknowledge the reception of that incorrect packet. After the timeout value, the transmitter retransmits the packet, again numbered 0, which arrives correctly. Transmission errors corrupt the acknowledgment.

trated in Figure 2.17. The receiver gets the packets in order and needs only to keep track of the number (0 or 1) of the last correctly received packet to identify duplicates that may arrive when acknowledgments are lost or late. Figure 2.18 shows that the transmitter must number packets. In the situation of the figure, if the packets were not numbered, the receiver could not tell whether the second packet it receives is a copy of the first or a new packet.

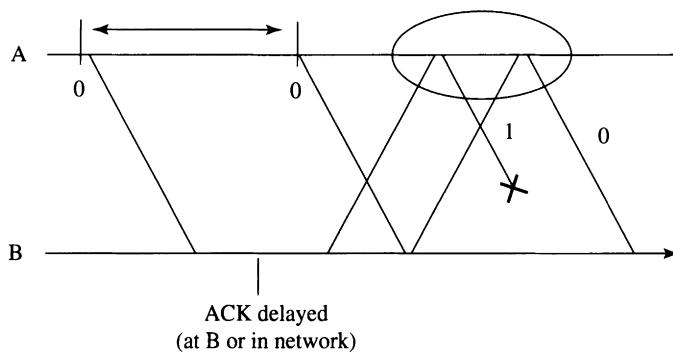
Figure 2.19 shows why the receiver must number the acknowledgments. The figure assumes that the acknowledgments are not numbered and shows a sequence of events that lead the protocol to fail to retransmit a packet. The figure shows that the transmitter can be led to confuse the acknowledgment of a packet 0 with the acknowledgment of a packet 1. To prevent such confusion, it is necessary for the receiver to number the acknowledgments.



2.18

FIGURE

In the left-hand part of the figure, a packet is correctly transmitted but its acknowledgment is not. Consequently, the transmitter sends a copy of the packet. In the right-hand part of the figure, the packet and its acknowledgment are correctly received so that the transmitter sends a new packet. The receiver could not distinguish these two cases if the packets were not numbered.

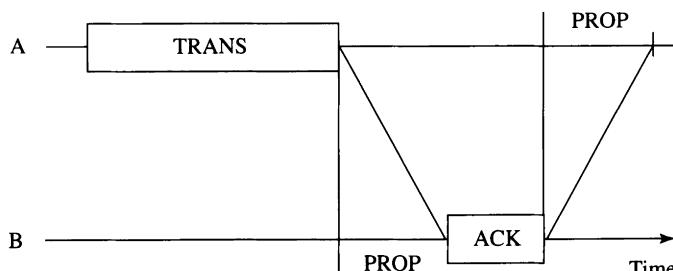


2.19

**FIGURE**

In the sequence of events shown here, the protocol fails to transmit a packet if the acknowledgments are not numbered.

The ABP protocol is inefficient if the propagation time is long compared with the transmission time, because the sender waits for the acknowledgment before sending the next packet. The efficiency of the Alternating Bit Protocol is defined as the fraction of time that the transmitter transmits new packets. When the bit error rate is negligible, the efficiency is equal to  $\text{TRANS}/(\text{TRANS} + 2\text{PROP} + \text{ACK})$ . (See Figure 2.20.) In this expression, TRANS denotes the time to transmit a packet, ACK the time to transmit an acknowledgment, and PROP the time taken for the signal to travel from the



$$\text{Efficiency} = \frac{\text{TRANS}}{\text{TRANS} + \text{ACK} + 2\text{PROP}}$$

2.20

**FIGURE**

The figure shows the time that the transmitter takes to transmit a packet and get its acknowledgment.

sender to the receiver and from the receiver to the sender. (We assume these two times to be equal.)

### *Go Back N*

A protocol that is more efficient than ABP for long propagation times is Go Back  $N$ . The network designer or user selects a window size  $N$ . Typically,  $N$  is just large enough so that the pipe is full: the sender gets the acknowledgment of the first packet when it finishes transmitting packet number  $N$ . The sender numbers the packets sequentially modulo  $N + 1$ , that is,  $1, 2, \dots, N, 1, 2, \dots, N, \dots$

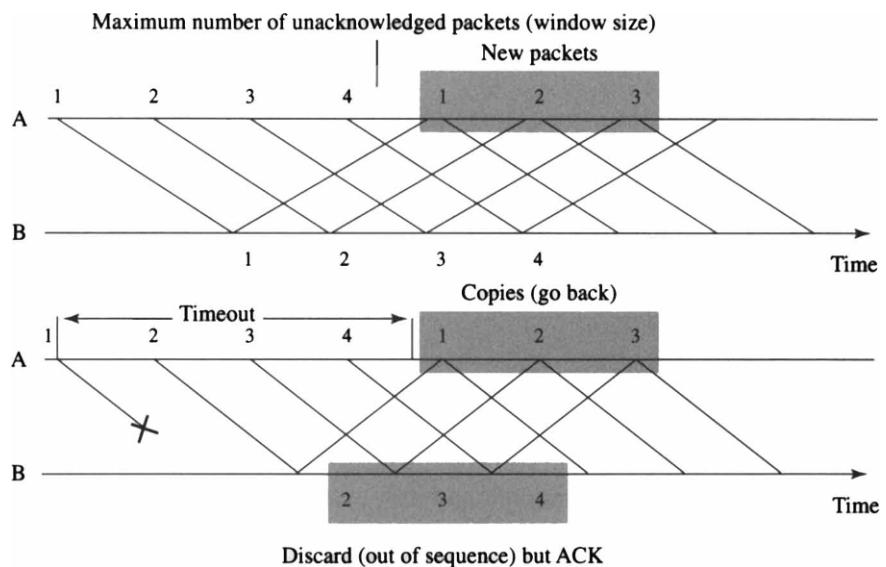
The receiver acknowledges every correct packet that it receives with an acknowledgment that it numbers as the largest numbered packet that it has received so far. For instance, if the receiver gets the packets  $1, 2, 4, 5$ , then it sends the acknowledgments with numbers  $1, 2, 2, 2$ . Before transmitting packet number  $n + N$  (modulo  $N + 1$ ), the sender waits until it receives the acknowledgment of packet number  $n$ . Whenever the sender sends a packet, it starts a timer with a timeout  $T$  selected by the network designer. When the sender fails to receive the acknowledgment of a packet before the packet times out, the sender retransmits that packet and all the packets that it has transmitted since it last transmitted that packet. Therefore, the receiver need not store any packet since it eventually gets them correctly in sequence. The sender must be able to store up to  $N$  packets. Note that with this protocol the sender may have to retransmit packets that were correctly received by the receiver. Figure 2.21 illustrates the sequence of events in Go Back  $N$  with  $N = 4$ .

In Figure 2.22 we illustrate the calculation of the efficiency of Go Back  $N$  where there is no transmission error. The efficiency is given by

$$\text{Efficiency} = \min \left\{ \frac{N \text{ TRANS}}{\text{TRANS} + \text{ACK} + 2\text{PROP}}, 1 \right\}.$$

The efficiency increases linearly with  $N$  up to 100%. To appreciate the magnitudes that may be involved in practice, we consider a numerical example. Suppose the packet size is 2,000 bits, the ACK is 80 bits, transmission is over a 30-km fiber link at 155 Mbps. Then

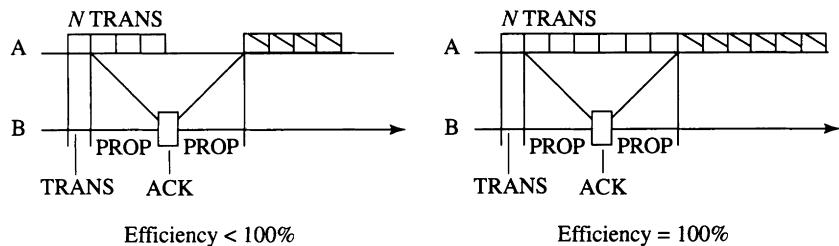
$$\begin{aligned} \text{Efficiency} &= \min \left\{ \frac{N \text{ TRANS}}{\text{TRANS} + \text{ACK} + 2\text{PROP}}, 1 \right\} \\ &= \min \left\{ \frac{N \times 2000}{2000 + 80 + 46,500}, 1 \right\}. \end{aligned}$$



2.21

FIGURE

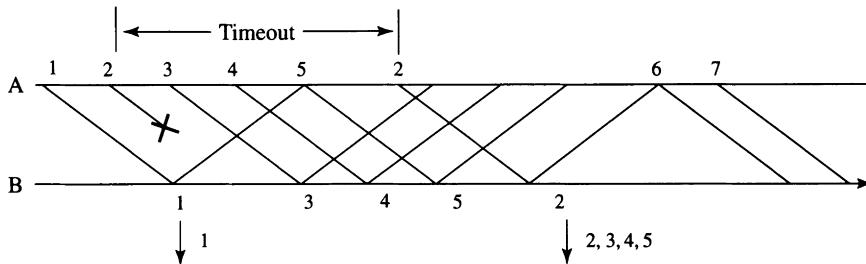
The top panel shows the sequence of transmissions using Go Back N with a window size of four with no transmission errors. The bottom panel shows the “go back” being triggered when the transmitter fails to get an acknowledgment of packet 1 before the timeout.



2.22

FIGURE

The transmitter can send  $N$  packets before it gets the acknowledgment of the first one. Consequently, when no errors corrupt the transmissions, the packets are transmitted in cycles of  $N$  transmissions. That observation enables us to calculate the efficiency in the text.



2.23

FIGURE

When SRP is used, the receiver can accept and store packets out of sequence. The transmitter retransmits only the packet for which it fails to get an acknowledgment before a timeout.

So the efficiency is 4.2% when  $N = 1$ , and 100% for  $N \geq 24$ . The network engineer should propose a window size equal to 24 in this example. Note that Go Back  $N$  with  $N = 1$  is exactly ABP. Transmission errors reduce the efficiency of Go Back  $N$ . The analysis of the efficiency when errors occur is more complicated.

### Selective Repeat Protocol

The final protocol that we explain is the *Selective Repeat Protocol* (SRP). SRP is similar to Go Back  $N$  but retransmits only packets that were not correctly received. A window size  $2N$  is agreed on, and the packets and acknowledgments are numbered modulo  $2N$ . Once again, the sender waits for the acknowledgment of packet  $n$  before sending packet  $n + N$  (modulo  $2N$ ). The sender retransmits packets that have not been acknowledged within a timeout after their transmission, and the receiver acknowledges all the correct packets. The sender and the receiver must both be able to store up to  $N$  packets. Figure 2.23 shows typical sequences of events when the nodes use SRP (with  $N = 4$ ).

#### 2.6.4 Flow Control

Consider a source that is sending packets to a receiver. The receiver sets aside some memory space to store the incoming packets until it can process them, possibly by displaying them or storing them on a hard disk. When the memory gets full, the receiver should notify the source to stop transmitting.

When the source uses a window mechanism, the receiver can indicate, in the acknowledgments, the maximum window size that it can store. The source

then uses that value as a limit to its window size. As the receiver buffer becomes full, this maximum value decreases, which slows down the source and prevents overflowing the receiver memory.

### 2.6.5 Congestion Control

The bit streams of many sources get multiplexed and read into a switch buffer. If the sources are unregulated, occasionally the buffer will get filled up, causing long delays and, possibly, buffer overflow and resulting packet loss. Congestion-control mechanisms can cause the sources or switches to reduce the number of packets they inject into the network when conditions of congestion are detected.

If the sources use a window mechanism as in Go Back  $N$  or SRP, then this mechanism can be adapted to serve the purposes of congestion control. Notice that the sources can detect congestion or packet loss from the fact that their packets time out. Thus as soon as several timeouts occur in quick succession, the source should reduce the window size. This reduction automatically reduces the number of outstanding packets, and hence network congestion.

There is a conflict between keeping the window size small to reduce congestion and keeping it large enough to “fill the pipe” to maintain high efficiency. Imagine a transmission between two hosts and assume that the round-trip time between the hosts is  $T$  seconds and the transmission rate is  $R$  bits per second. The “delay-bandwidth” product of the connection is defined as  $R \times T$ . For instance, for a short wireless link, possible values might be  $R = 30$  Kbps and  $T = 10\mu\text{s}$ , so that  $R \times T = 0.3$  bit. As another example, for a fast cross-country backbone link, the values might be  $T = 30$  ms and  $R = 2.4$  Gbps, so that  $R \times T = 72$  Mb.

Suppose the delay-bandwidth product is large and the window size is large. Then most of the outstanding packets are propagating through the links, rather than waiting in the buffers. As a result, once congestion is detected and the window size is reduced, the many packets already in transit in the links are unaffected by the reduced window size. Those packets continue to arrive into the buffers, increasing congestion and packet loss. Thus window flow control is not effective when individual connections have a large bandwidth-delay product. In that case, “open loop” congestion control in the form of so-called rate control is more effective. We study these techniques in detail in Chapters 8 and 9.

## 2.6.6 Resource Allocation

Different applications require bearer services of different qualities (delay, error rate, etc.). A network can guarantee an application a particular service quality only if it can dedicate resources (bandwidth, buffers) to that application.

A circuit-switched network dedicates a fixed bandwidth to each connection, so it can guarantee minimum delay to an application whose peak rate is less than the fixed bandwidth.

A datagram network is connectionless, and the network switches do not have the state information needed to distinguish packets from different applications. As a result, the network cannot dedicate resources that are specific to individual applications. Some crude resource allocation is possible, however. For example, as will be seen in Chapter 4, the TCP/IP protocol permits expedited packets to be treated with a higher priority than nonexpedited packets. As another example, switches can provide differentiated services to packets of different types by implementing a priority or weighted scheduling.

In virtual circuit-switched networks, each packet carries its virtual circuit identifier or VCI. This allows switches to treat packets differently based on their VCI. Thus, at the time its virtual circuit is set up, an application can negotiate with the network for certain service quality. The network switches can then reserve bandwidth and buffers for that virtual circuit so as to provide that quality. Guaranteed service quality is thus a possibility. In Chapters 8 and 9 we consider resource allocation mechanisms that provide this guarantee.

Table 2.2 summarizes the main techniques used to implement the basic network operations.

Function	Technique	Characteristics	Reference
Multiplexing	Time division (TDM)	Fixed bit rate channels, flexible, minimum overhead to identify channels	Ch. 5
	Frequency division (FDM)	Fixed bandwidth channels, very flexible, suited for analog signals	
	Statistical (SM)	Channel bandwidth on demand, overhead for channel identifier	
Switching	Circuit (CS)	Good for CBR traffic, no queuing delay, low utilization for bursty traffic	

(continued)

Function	Technique	Characteristics	Reference
	Packet (PS)	Good for connectionless message traffic, variable queuing delay, high utilization for bursty traffic, robust against link failures	Ch. 3
	Virtual circuit (VC)	Good for connection-oriented VBR traffic, variable queuing delay, good utilization	Ch. 6, 8
Error control	Detection	CRC codes can detect most errors	
	Retransmission	Alternating Bit, Go Back N, etc., used to control errors when reverse path is available	
	Error correction	Used when reverse path is not available, e.g., storage or high-delay satellite links	
Flow control	Window flow control	Combined with end-to-end error control protocol, effective in low delay-bandwidth connections	Ch. 8, 9
	Rate control	Effective in high delay-bandwidth connections, used in ATM services	Ch. 8, 9
Resource allocation	Routing, bandwidth, buffer allocation; admission control	Used in virtual circuit networks to guarantee service quality	Ch. 8, 9

2.2

TABLE

Techniques and characteristics of basic network operations.

## 2.7

## LAYERED ARCHITECTURE

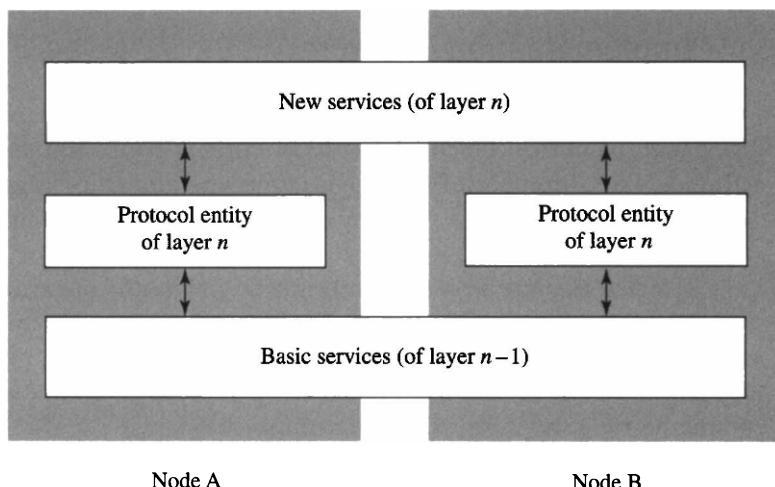
An *architecture* is a specific way of organizing the many functions performed by a computer network when it provides services such as file transfer, e-mail, directory services, and terminal emulation.

In this section we first introduce the *layered* architecture of network functions. We then comment on the implementation of layers.

### 2.7.1 Layers

In most networks, the functions are organized into *layers*. As shown in Figure 2.24, in a layered decomposition, services of layer  $n$  are implemented by protocol entities (processes) at layer  $n$  using the services of layer  $n - 1$ . Examples of such a layered decomposition of communication functions abound in everyday situations. For instance, two executives can exchange messages by using the services of their secretaries. The secretaries themselves exchange messages by using facsimile machines. The machines transmit facsimiles by using the services of the telephone network. (See Figure 2.25.) We covered a more technical example when we explained error control: by using some suitable protocol, a transmitter and a receiver can implement reliable packet transmissions over an unreliable transmission link.

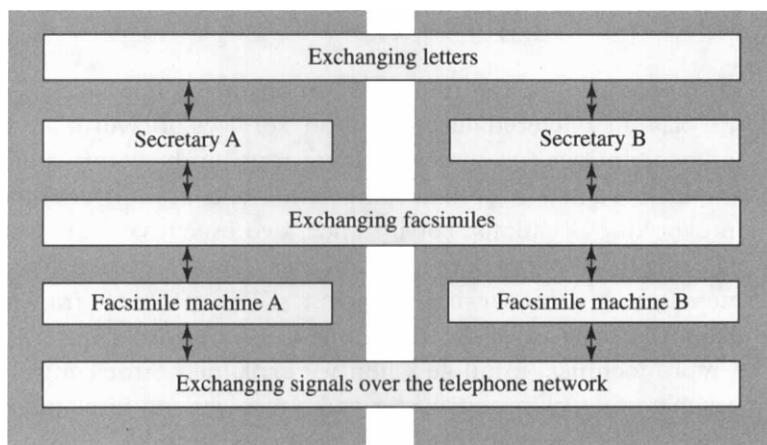
By decomposing network functions into layers, the network engineers partition a complex design problem into a number of more manageable subproblems—those of designing the different layers. This decomposition simplifies the design and its verification. Moreover, the decomposition permits standardization, makes possible the competitive implementations of different layers, and facilitates network interconnection. For the layered decomposition to provide these benefits, the network engineers must specify without ambiguity the services provided by different layers and the interfaces between



2.24

FIGURE

In a layered architecture, protocol entities of layer  $n$  implement services by using the services implemented by layer  $n - 1$ .



**FIGURE 2.25** This figure illustrates that we can view the exchange of letters by secretaries using facsimile machines as a multilayered process.

layers. Standardization bodies such as the ITU (International Telecommunication Union), ISO (International Organization for Standardization), IEEE (Institute of Electrical and Electronics Engineers), and ANSI (American National Standards Institute) organize working groups that develop and publish these specifications. The resulting specifications, called *standards*, are necessarily detailed and lengthy.

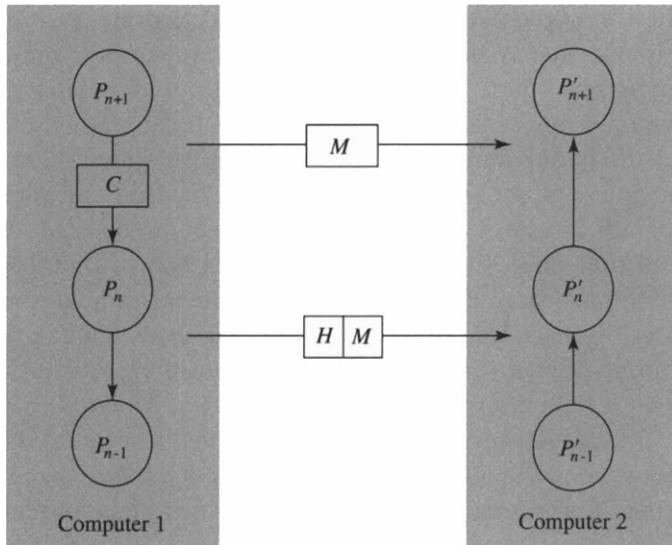
### 2.7.2 Implementation of Layers

When protocols are arranged into layers, the protocol entities of adjacent layers exchange messages. Since the protocol entities are computer processes, this implies that a number of processes communicate. Once it gets a message, a protocol entity performs some operations before it transmits the message to the next protocol entity.

To make the interprocess communication and the operations of a protocol entity more concrete, we examine how these functions are implemented. Our main point is to show the actual message passing between processes that the layered models represent. We also discuss the performance implications of implementing the interprocess communication.

Our discussion does not cover all possible implementations nor all possible types of operations. However, it captures some of the main features of implementations.

Our objective is to explain the steps shown in Figure 2.26. The figure shows protocol entities in two computers. The protocol entity  $P_{n+1}$  at layer  $n + 1$  in



2.26  
FIGURE

The protocol entity  $P_{n+1}$  asks layer  $n$  to transmit a message  $M$  to  $P'_{n+1}$  by sending a control message  $C$  to  $P_n$ . The entities  $P_n$  and  $P'_n$  execute the protocol of layer  $n$ . The text describes implementations of the message passing between  $P_{n+1}$  and  $P_n$  and the execution of the protocol by  $P_n$  and  $P'_n$ .

computer 1 sends a message  $M$  to the peer entity  $P'_{n+1}$  in computer 2. To send that message,  $P_{n+1}$  sends a control message  $C$  to the protocol entity  $P_n$ , asking it to transmit  $M$  to  $P'_n$ . The entity  $P_n$  adds a header  $H$  to the message  $M$  before it sends it to  $P'_n$ . This header may contain addresses, sequence numbers, control fields, and error-detection bits that  $P_n$  and  $P'_n$  need to supervise the transmission of  $M$ .

We first examine how  $P_{n+1}$  passes the message  $C$  to  $P_n$ . This message passing can be implemented by using a shared memory or by using a queue.

When using the shared memory method, the processes  $P_{n+1}$  and  $P_n$  have access to a common memory segment that is divided into  $N$  locations that store data and to a common variable  $X$ , called a *semaphore*, that can take the values  $0, 1, \dots, N$ . The variable  $X$  represents the number of locations that are available to be written. Process  $P_{n+1}$  can write as long as  $X > 0$ , and it decrements  $X$  by 1 whenever it has written into a location. Process  $P_n$  can read whenever  $X < N$ , and it increments  $X$  after it has read a location. Semaphores can also be used by multiple writer and reader processes. When this possibility is implemented, special care must be taken to avoid conflicting manipulations of the semaphore value; likewise, caution is called for when handling situations in which a process aborts before the semaphore has been reset.

When the processes use queues to communicate, process  $P_{n+1}$  writes  $C$  into a queue that is read by process  $P_n$ . A queue is organized as a first-in, first-out array of data. The queue has some reserved capacity, and it can be implemented by the operating system as a linked list with a pointer to the head of the queue and another to the tail of the queue. Process  $P_{n+1}$  writes into the queue (at the tail), and process  $P_n$  reads from the queue (at the head). The operating system checks that there is data to be read when  $P_n$  wants to read and that there is space available when  $P_{n+1}$  wants to write. Typically, the operating system can handle a large number of queues between various processes by sharing a large memory among these queues. The capacities of the different queues can be adjusted dynamically by creating a new linked list whenever a new interprocess queue is needed. The different queues can be used to pass messages that should be handled differently. For instance, one queue may contain high-priority messages and another low-priority messages. A number of processes may write into the same queue or read from the same queue if, for instance, each message in the queue contains an identification number that specifies the process for which it is intended.

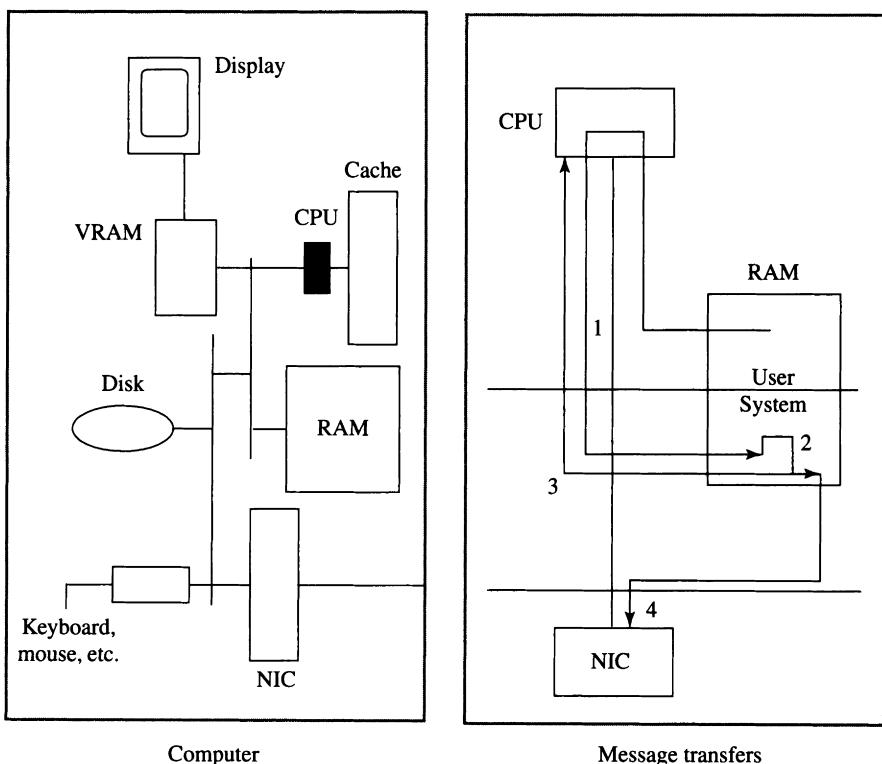
Typically,  $P_{n+1}$  does not transmit the message  $M$  to  $P_n$ . Instead, it transmits a pointer to that message that indicates where  $M$  is in memory and its length. The actual transfer of  $M$  must occur when the message must move across different computer boards. In many systems, the network interface board has its own memory that stores the packets ready to be transmitted. In such an implementation, it may be that  $P_n$  is implemented by the network interface board while  $P_{n+1}$  is implemented by the main CPU. In that case,  $P_{n+1}$  actually copies  $M$  to  $P_n$  across the computer bus to which the interface board is attached. Obviously, the message must also be copied by the bottom layer, which implements the actual transmission between computers.

Let us now turn to the implementation of the functions that  $P_n$  performs. The entities  $P_n$  and  $P'_n$  implement the protocol of layer  $n$ . This protocol specifies that  $P_n$  must compute the header  $H$ , start some timer, and update some counters. When a timer expires,  $P_n$  typically initiates a new call to  $P_{n-1}$  to retransmit the message. The entity  $P'_n$  must read the header  $H$  and perform a set of operations such as verifying that the packet is correct, send an acknowledgment, and indicate to  $P'_{n+1}$  that a packet has arrived.

In some protocols, the header  $H$  contains an error-detection field whose value depends on the message  $M$ . In that case, to calculate  $H$ , the process  $P_n$  must read the message  $M$ , which, together with the calculation of  $H$ , requires a large number of instructions. In other protocols, the header  $H$  contains an error-detection field that is independent of  $M$  and that protects only the header itself. The execution of such a protocol is typically much faster.

This discussion points to the implications of both the design and the implementation of protocols for the achievable rates of execution of such protocols. Fast protocols are designed to limit the need for protocol entities to read full messages. Protocol implementations are faster when they minimize the number of actual message transfers by passing pointer values instead of copying the messages. Finally, protocol executions can be speeded up by implementing protocol entities on dedicated hardware that frees up the main CPU. Ideally, the execution of the protocols should impose a minimum burden on the main CPU, and it should be fast enough to keep up with the communication link and with the source of the data to be transferred.

We make this discussion more concrete with the help of Figure 2.27. The panel on the left shows a basic host computer architecture. The CPU and the



2.27  
The left panel gives a simple architecture of a host computer and its connection to the network. The right panel shows that four copies may be involved across the CPU bus to run an application, reducing the host throughput.

main memory communicate over the CPU bus; there also is an I/O bus for communication with the network. A dedicated Network Interface Card (NIC) implements many of the functions dealing with the physical transmission and reception of packets. The panel on the right shows that four copies of a file across the CPU bus are needed when the file is transferred by FTP (File Transfer Protocol), studied in Chapter 4. The file, initially in user space in memory, is first copied into system space, where it is handled by FTP (1). Then FTP copies the file over to the next layer protocol TCP (Transmission Control Protocol) (2). The TCP protocol entity fragments the file into IP (Internet Protocol) packets. The CPU now computes the CRC bits of each packet, which requires a third copy (3). Finally, each completed IP packet is forwarded to the NIC (4). The NIC transmits the packet over the network. If the CPU bus has a throughput of 320 megabytes per second (MBps) (80-MHz clock and 4-byte-wide bus), the four copies have reduced this to 80 MBps.

We have introduced the concept of layered architectures. We now describe a very useful architecture, the Open Data Network or ODN model.

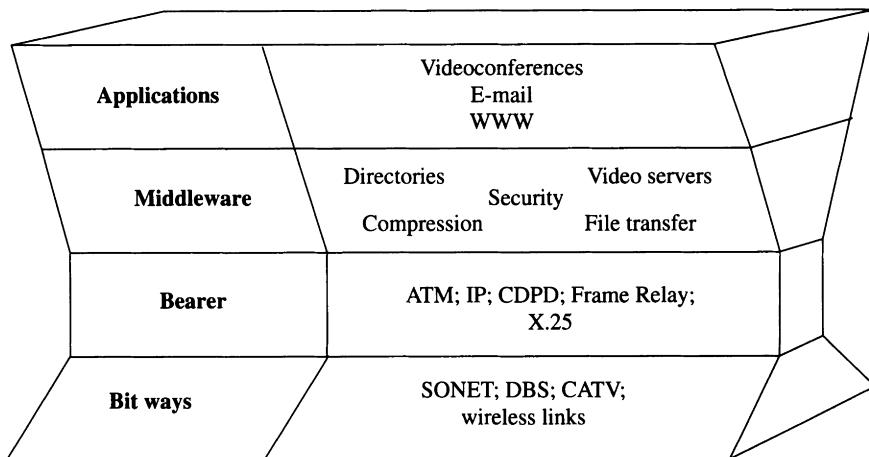
## **2.8 OPEN DATA NETWORK MODEL**

The Open Data Network or ODN model was recently proposed by a panel of network engineers as a framework within which the telephone, computer, and CATV networks can be located and compared. The purpose of the ODN model is not to develop a standard like the OSI model studied in Chapter 3, but to help understand how it may be possible to interconnect these three types of networks, despite the differences in their technologies, services, and markets.

The ODN model is displayed in Figure 2.28. It has four layers, called *bit ways*, *bearer*, *middleware*, and *applications*, as shown on the left side of the figure. Examples of implementations of those layers in specific networks are listed on the right.

The service provided by a *bit way* is the transport of bit streams over a link. The bit way may be implemented by a SONET link of the telephone network, by a direct broadcast satellite or DBS link, by a CATV link from the head station to a user, by a cellular radio channel, or by some other wireless connection. The bit way provided by a specific link technology can be characterized by its speed, delay, and error rate.

A *bearer* service is the end-to-end transport of bit streams in specific formats. For example, in the ATM bearer service, 53-byte cells are transported



2.28

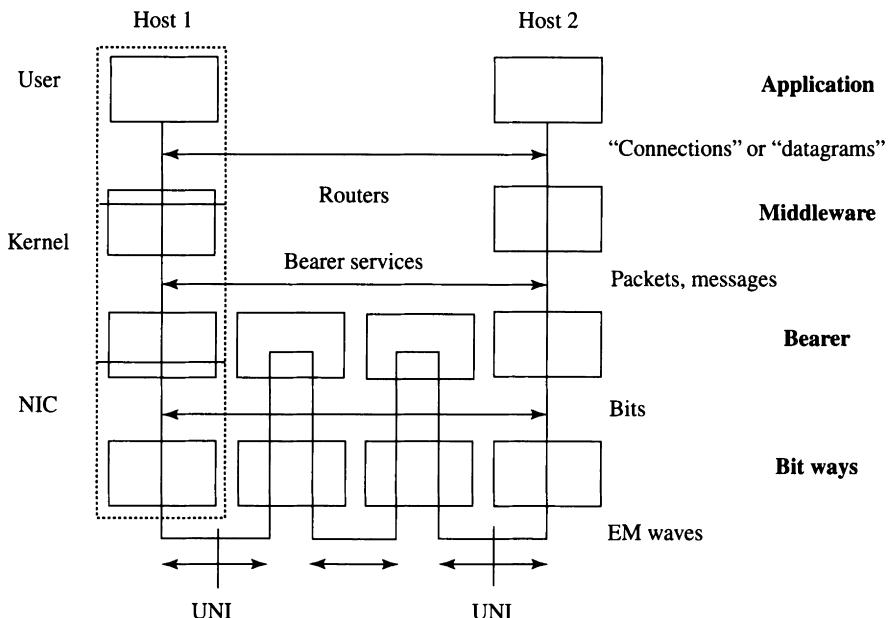
The Open Data Network model has four layers.

FIGURE

end-to-end over virtual circuits. The format of the cells is fixed. In the network or IP layer of the Internet, the bearer service transports variable-sized datagrams in a specified format from source to destination over a packet-switched network. As we will see, the bearer services layer is the most important layer from the viewpoint of network interconnectivity.

The *middleware* services are generic services that are used by a large number of applications. Examples of such middleware services are file transfer, directories, and video servers. These services could be provided by individual user computers. This is indicated in Figure 2.29, which shows middleware services being provided by the operating system. In a campus environment, however, economies of scale motivate the installation of dedicated servers for directories and other databases, software “warehouses,” and the like.

For some middleware services, the economies of scale are even greater, and this may justify installing special hardware and software in network nodes to provide those services. For example, cellular phone standards like GSM and IS-54 compress a voice signal into a 16-Kbps bit stream. The telephone company switches contain hardware that decompresses those bit streams to recover the original voice signal. (If this were not done, only telephone sets equipped with decompression hardware could receive the compressed signal.) Similarly, CATV networks may transport a compressed video signal to the curbside where it is decompressed before distribution to individual users.



2.29

FIGURE

Networks with the same bearer services can be connected via routers. The simpler the bearer services, the easier the network interconnection.

Finally, the *application* layer provides the services that users want. Examples include e-mail, WWW (World Wide Web), video on demand, and video-conferencing. As indicated in Figure 2.29, the application layer is usually implemented in the host computer. This often involves proprietary software, such as Eudora for e-mail and Intuit's Quicken for home banking. Sometimes specialized hardware is needed to run the application. Thus, "pay TV" requires a "set-top box" that unscrambles or decompresses a TV signal. The set-top box helps create a record for billing purposes. Because applications often involve proprietary and expensive "program content" (e.g., a movie) as well as transport services, provision must be made for billing for this program content.

Much profit will be made by hardware and software manufacturers whose set-top boxes become a *de facto* standard. If a manufacturer's equipment is widely adopted by users, program content providers will have an incentive to conform to that equipment, and the equipment manufacturer can then extract a monopoly rent from the content providers for use of that equipment. In 1995 the contenders for this set-top "prize" included the large software

companies, such as Microsoft and Sybase, and the telephone companies. On the other side, companies such as Sun Microsystems, which have not yet developed competitive products, are lobbying for government regulation that will maintain an “open” application layer.

We can now see how different layers cooperate to produce sophisticated user services. An application may make use of middleware services such as file transfer. File transfer service in turn is implemented using a bearer service such as IP datagram transport. And datagram transport is implemented using a bit way provided by a local area network such as Ethernet.

Finally, we study network interconnectivity. Observe that the ODN model of Figure 2.28 has a narrow waist at the bearer service layer. This is intended to suggest two things. First, a very small set of bearer services can be provided by a large variety of bit way implementations. Second, the small set of bearer services is sufficiently versatile to support a large variety of applications.

The small set of bearer services greatly promotes internetworking, as can be seen with the help of Figure 2.29. The figure shows two hosts, Host 1 and Host 2, belonging to two separate networks. These two networks can be interconnected by a router or switch, provided both networks support the same bearer service. (The router is attached to the bit ways of both networks.) The application running on Host 1 produces a bit stream that is recovered by the router using the bearer services of the first network. The router then forwards that bit stream to Host 2 using the bearer services of the second network.

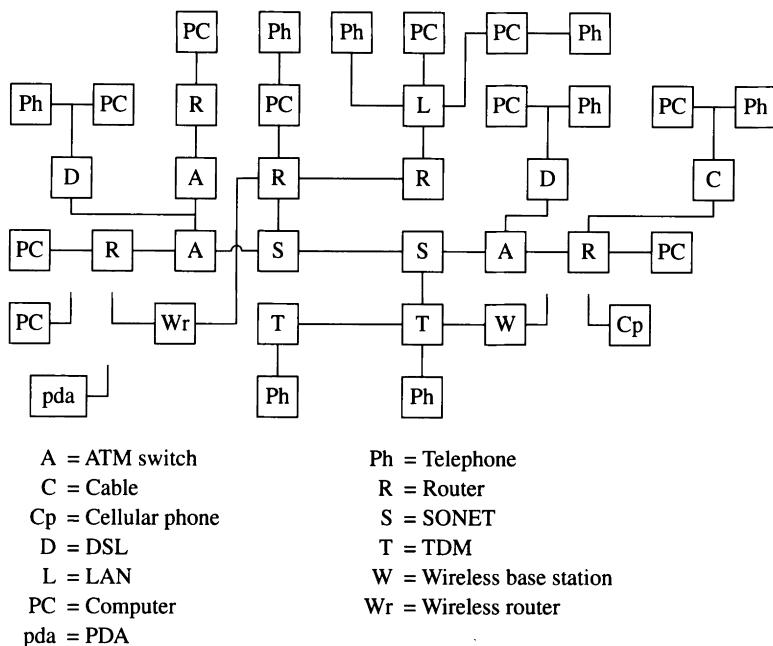
We thus see that the simpler the set of bearer services, the easier it will be to interconnect networks. The most important examples of this are the Internet Protocol (IP) and ATM, both of which specify a single bearer service. On the other hand, the set of bearer services should be versatile enough to support a wide range of applications. We will see that in this respect ATM is better than IP.

---

## 2.9 NETWORK ARCHITECTURES

In this section we discuss network architectures. Our objective is to explain the interoperability of the different technologies.

Figure 2.30 shows a collection of telephone sets, computers, and personal digital assistants (PDAs). (Television sets would fit in this collection as well.) These devices are interconnected with links that are wireless, copper pairs, coaxial cables, or optical fibers. These links transport data and voice (possibly video) either as bits in TDM frames, or in ATM cells, or in IP packets.



2.30

FIGURE

Networks combine many different technologies. The power of the TCP/IP protocols is in enabling their interoperability on a large scale.

The local telephone lines use a TDM technology that is not globally synchronized. Long-distance telephone lines use the synchronized technology over fibers (SONET).

ATM cells can be transported over copper lines either directly if the lines are short or using DSL modems if they are longer. ATM cells can also be transported over optical links. When the fiber is short, a simple transmission scheme can be used. When the fiber is longer, one can use the SONET framing, clock recovery, and error detection, without having to synchronize the link. This use of SONET, sometimes called *SONET-Light*, is becoming popular in 1999. If the transmission is over a long distance, the ATM cells can be sent over a SONET "circuit" (the proper term is *SONET path*) to the next ATM switch. This combination is then called *ATM over SONET*.

IP packets can be transported between PCs and routers over a LAN, over a cable (equipped with cable modems), over wireless links, or over optical fibers (using SONET-Light or SONET). ATM cells can also transport the IP packets.

All these combinations can seem excessively complicated. Their coexistence is explained by the need to support different “legacy” systems. For instance, SONET was developed to make the long-distance telephone equipment simpler. Since this long-distance digital network is available, it makes sense to use it to transport ATM cells and IP packets. On the other hand, IP networks in the form of intranets and the Internet are somewhat capable of transmitting voice. One then ends up with voice over IP over ATM over SONET, which is indeed a “baroque monstrosity,” to quote a respected IP researcher. We discuss the plausible evolution of this technology in Chapter 13.

## 2.10 NETWORK BOTTLENECKS

Before we embark on a detailed study of networks, we pause to comment on the current major technology bottlenecks in achieving a high-performance network. The rest of the book elaborates on the issues that we introduce in this brief section.

### *Is Bandwidth Plentiful and Free?*

The transmission rate of links is rapidly increasing. Optical links can transmit at 10 Gbps or more over about 100 km. Moreover, with wave-division multiplexing, or WDM, a single fiber can carry a large number (16 to 64 commercially, up to 512 in the laboratory) of such fast transmissions. Consequently, a transmission link with 1 Tbps (equal to  $10^{12}$  bps) is feasible between two cities. If one large city has one hundred thousand users that are simultaneously active, each user can in principle get a transmission rate of 10 Mbps, which is more than adequate, as we know from our experience with local area networks that transmit at such a rate. This simple discussion appears to justify a commonly held belief that bandwidth is or will soon be an abundant commodity.

A closer study of network operations reveals a more complex situation. Although the backbone network is built with optical fibers, many access links still use copper or shared coaxial cable whose transmission rates are comparatively slow. Another, more subtle source of difficulty is that many sources of data traffic are greedy and try to fill up the available bandwidth, as we explain when we study TCP. Consequently, increasing the bandwidth does not eliminate the competition for it nor the delays and losses it entails. Moreover, in order to exploit the vast transmission rate of optical links in the backbone, switches,

routing protocols, transport protocols, and applications must be improved. The main objective of this book is to explain these improvements.

### ***Switch Modifications***

The switch must make a forwarding decision for each packet that arrives and must forward the packet to the corresponding output port. Better look-up algorithms can speed up the forwarding decision. New switch architectures speed up the forwarding of packets. Different applications have vastly different delay, throughput, and loss requirements. Switches are designed that can handle packets according to these requirements.

### ***New Routing Protocols***

To simplify the task of the switches, new protocols are being designed. These protocols label the packets according to the requirements of the application and possibly based on some precomputed routing decisions. It is simpler for the switch to use the label than the destination address and an application identification to determine how it should handle the packet. As the number of subscribers to “push” programs increases, multicast routing may become an important method to improve the efficiency of the routing.

### ***Transport Protocols***

The end hosts (the source and the destination) control the pacing of packet transmissions. As the links get faster, the number of packets in transit increases, which modifies the dynamics of the control mechanisms. The control mechanisms probably need to be modified to match these changing operating conditions.

### ***Applications***

Faster networks make new applications possible. Whereas a few years ago most of the applications were text-based, today's applications involve multimedia. As the network speed continues to increase, we will see more conversation applications and Web sites with richer content. More people will use live radio and video programs. Telecommuting and residential videoconferencing may become more commonplace.

It may be that for some time to come, these new applications will maintain their pace of increasing the traffic to the capacity of the network. If this prediction holds true, then optimization of resource utilization will remain critical. Pricing of services may become necessary to protect essential applications.

**2.11****SUMMARY**

Networks provide communication services needed to support user applications. These services are provided from more elementary services in a layered architecture, like the Open Data Network model. Some of the layers are implemented in the network switches, others in host computers. The network itself provides bearer services, that is, the end-to-end transport of bit streams. The bearer services are implemented by network links and switches using mechanisms of multiplexing, switching, error control, flow control, and resource allocation.

Applications generate constant or variable bit rate traffic or message exchanges. The applications impose certain requirements on the bearer services that transport this traffic. Those requirements are expressed in terms of bandwidth, delay, and error rates. Circuit switching meets the most stringent requirements in terms of delay and bandwidth but may lead to such poor utilization that it becomes uneconomical. Datagram switching is the most efficient, but it may not be able to provide guaranteed delay or bandwidth. Virtual circuit switching can combine high utilization with the ability to meet guarantees in delay and bandwidth.

In Chapter 3 we study packet switching, in Chapter 5 we study circuit switching, and in Chapter 6 we study ATM networks, the most important type of virtual circuit switching network.

**2.12****NOTES**

Shannon's theorems, which form the basis of information theory, are discussed in many texts; see [CT91].

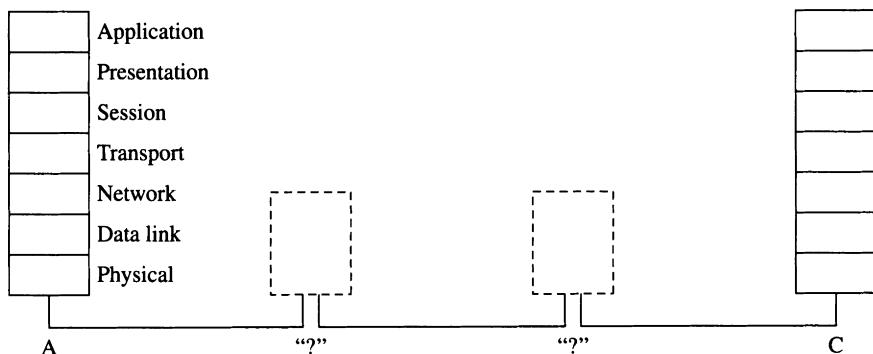
The notion of layered architecture, modularity, and hierarchy shows up in various parts of computer science as well as in communication networks; see [T88, W98]. For details on implementation of protocols in UNIX, see [P93]. The Open Data Network model appears in [Kle94].

**2.13****PROBLEMS**

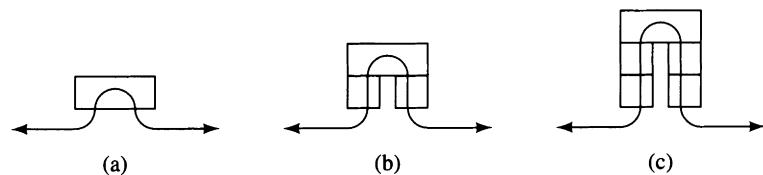
1. It is very expensive to store and archive X rays for medical diagnosis. The current system uses large photographs. An electronic image of comparable

quality would require a display of  $1,000 \times 1,000$  pixels, with a 16- or 24-bit grayscale, and with a zoom-in capability. How expensive is a monitor capable of displaying so much information? (How many pixels does your monitor display, and how many grayscale levels does it permit?) If a radiologist is retrieving the X ray from an archive, the acceptable delay is 10 s, say. What should be the bit rate of the links connecting the archive to the radiologist's office? If the X rays that the radiologist is going to view are known, say 10 min in advance, one could retrieve the X rays early and buffer them locally. This permits a reduction in the bit rate at the cost of increasing local storage. What is the bit rate/buffer trade-off? Suppose the cost of increasing the bit rate by 1 Mbps is  $r$  times the cost of 1 Mbps of disk. For what values of  $r$  is it worth reducing the link rate and increasing local disk storage?

2. The figure below illustrates the OSI seven-layer model for communication between nodes A and C. The two intermediate "?" nodes are also shown.



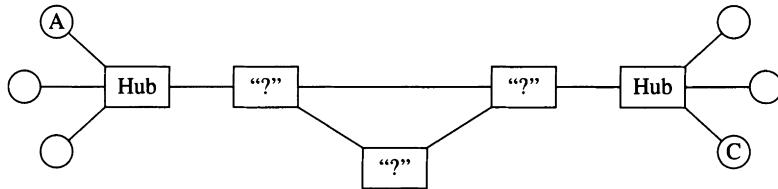
The next figure shows three possibilities for the "?" nodes.



What kind of device is shown in (a), (b), and (c) of this figure?

3. (a) Consider the network topology depicted in the figure on the next page where the boxes labeled with "?" are Ethernet hubs.  
  - (1) Is this a valid topology? Why or why not?

- (2) If this is a valid topology, can packets ever circulate around the loop?
- (3) If packets ever do circulate around the loop, what keeps them from circulating forever?
- (b) Assume that the boxes labeled with "?" are **Ethernet switches**. Answer the same sub-questions from (a).
- (c) Assume that the boxes labeled with "?" are **Internet Protocol routers**. Answer the same sub-questions from (a).



4. As seen in Figure 2.1, we may represent the interconnection of network elements as a network graph whose edges are the links and whose vertices are the switches and user nodes. Formally, we represent the network as a graph  $G = (V, E)$  where  $V = \{1, 2, \dots, N\}$  is the set of vertices and  $E \subset V \times V$  is the set of edges. The interpretation is that  $(i, j) \in E$  if there is a one-way transmission link from switch  $i$  to switch  $j$ . This representation is useful to specify and verify many network algorithms.

Suppose there is a cost  $C_{ij} > 0$  associated with each link  $(i, j) \in E$  representing delay or dollar cost of transmitting one packet over that link. We assume that the knowledge of  $C_{ij}$  is *local*, that is, router  $i$  knows only the costs  $C_{ij}$  of links that originate at  $i$ .

The problem is to build a shortest-path routing table at each router. The table at  $i$  has the following form:

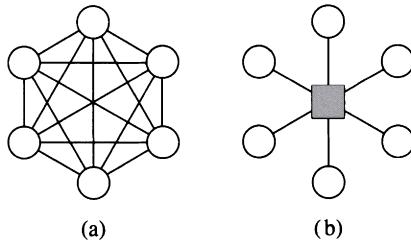
Destination router	Outgoing link	Cost to destination
1	$j_1$	$D_{i1}$
2	$j_2$	$D_{i2}$
$\vdots$	$\vdots$	$\vdots$
$N$	$j_N$	$D_{iN}$

The first row in the table is interpreted like this. If a packet for destination #1 arrives at router  $i$ , it should be forwarded over link  $(i, j_1)$ . The minimum

cost incurred by this packet from router  $i$  onward is  $D_{i1}$ . The other rows are interpreted similarly.

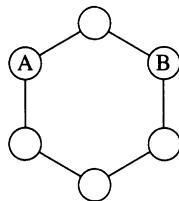
In order that each router be able to construct its own table, it will need to exchange some information with its neighbors.

- (a) Construct a distributed algorithm in which each router iterates over a two-phase cycle. In the first phase it updates its estimate of the table; in the second phase it communicates its table with its minimum cost estimate to its immediate upstream neighbors. How is the table initialized? Show that your algorithm converges after a finite number of iterations.
  - (b) Obtain an upper bound on the number of iterations needed for convergence.
  - (c) Is there any formal way in which you can say that the information that is exchanged between the routers is the *minimum* amount of information that must be exchanged?
5. The figure below shows two approaches for interconnecting nodes: the fully connected model and the shared-resource model.
- (a) If we use the fully connected model and we wish to connect  $n$  nodes, how many links will we require?
  - (b) What if we use the shared-resource model instead. Unfortunately we can't buy a single hub with  $n$  ports; all we can purchase are six-port hubs. Approximately, how many hubs and links will we require to connect  $n$  nodes?

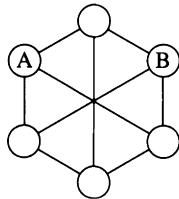


6. Consider the **ring topology** depicted in the figure at the top of the next page. This topology has the property that all nodes can still communicate even when there is a single link failure. Let's assume that the links in the network fail independently and with probability  $p$  in  $(0, 1)$ .
- (a) What is the probability that all nodes can communicate? Answer this part for the case of  $N = 6$ .

*Hint:* re-express this question in terms of the number of links that have failed.



- (b) Answer the previous part for any  $N$ .
  - (c) If we tell you that two of the links have failed, what is the probability that nodes A & B can communicate? Answer this part for the exact configuration shown in the figure (i.e.,  $N = 6$  and A & B located where indicated in the figure).
  - (d) Let  $D$  be the minimum distance path from A to B. What are the possible values for  $D$ ? What is the probability distribution of  $D$ ? Answer this part for the exact configuration shown in the figure.
7. Consider the network topology depicted in the figure below. Let's assume that the links in the network fail independently and with probability  $p$  in  $(0, 1)$ .
- (a) How many links may fail before network connectivity is lost?
  - (b) What is the probability of losing network connectivity? Answer this part with  $N = 6$ .
  - (c) Answer the previous part for any  $N$ .

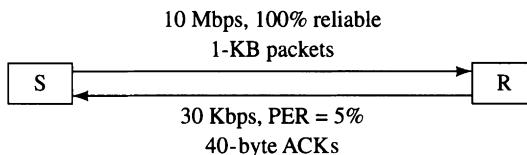


8. In this problem we consider the topologies of the previous two figures. Unlike the previous problems, the randomness here is in the choice of nodes A & B. (All links are assumed to be functioning.) Let  $N = 6$ . The nodes A & B are independently chosen at random and with equal probability. (i.e., as if we rolled a fair die to select A and then rolled again to select B.)
- (a) Consider the topology of the figure in Problem 6. Let  $D$  be the distance between nodes A & B. What is the expected value of  $D$ ?
  - (b) Repeat for the figure in Problem 7.

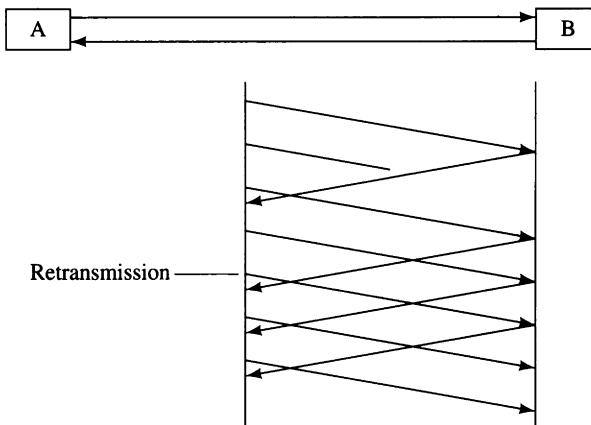
- (c) What part(s) of your solution to the first part were you able to reuse in the second part? Why?
9. What is the uncompressed bit rate for the NTSC TV signal? What compression ratio is achieved by MPEG1?
10. What is the propagation delay of a link from an earth station to a geostationary satellite? What would be the end-to-end delay of a voice conversation that is relayed via such a satellite?
11. A very common way to compress an audio signal is the following: Sample the audio signal. Denote the samples by  $x_0, x_1, \dots$ . Transmit  $x_0$ . Then transmit  $x_1 - x_0, x_2 - x_1, \dots$ . The dynamic range of the sample differences,  $x_i - x_{i-1}$ , is much smaller than the samples themselves, so the sample differences can be coded into fewer bits to achieve the same quantization noise. Develop a mathematical model that shows this.
12. Consider a video source that produces a periodic VBR stream with the following “on-off” structure. The source is “on” for 1 s with a rate of 20 Mbps; it is then “off” for 2 s with a bit rate of 1 Mbps.
- What are the peak and average rates of this source? Suppose this source is served at a constant bandwidth of  $c$  Mbps, where  $c$  is larger than the average rate. Calculate the buffer size  $b = b(c)$  in MB needed to prevent any loss as a function of  $c$ . What is the queuing delay as a function of  $c$ ?
  - If this traffic is produced by a videoconferencing application, which permits a maximum delay of 200 ms, what should  $c$  be?
  - If this traffic is produced by a video server that is downloading a one-hour-long video program, how much disk storage do you need? Suppose you want to play the program, but your disk access rate is only 10 Mbps. How many parallel disks would you need, and how would you store the video program on the disk?
  - In the description above, the period of the source is 3 s, and there is a duty cycle of 1/3. Consider another source with the same duty cycle but with a smaller period. Would you say the second source is more or less bursty? Why?
13. The analog phone access line has a bandwidth of 4 kHz. The line can be used to transmit digital voice at 64 Kbps or, using a modem, to transmit data at 9.6 or 11.4 Kbps. What is the spectral efficiency in each case? What kind of modulation scheme would you use to increase spectral efficiency?

14. Give examples of applications that can lead to multiplexing gain ranges displayed in Figure 2.7.
15. Explain why error-correction schemes are used (instead of error detection followed by retransmission) in data storage applications (such as audio CDs and magnetic disks) and in real-time applications (e.g., controlling a satellite).
16. A code with minimum distance 5 can
  - (a) correct up to 3 bit errors.
  - (b) detect up to 4 bit errors.
  - (c) correct up to 2 bit errors.
  - (d) detect up to 3 bit errors.
  - (e) detect up to 5 bit errors.
17. Packet switching is more efficient than circuit switching for bursty traffic because
  - (a) the switching delay is smaller for packet switches.
  - (b) packet switching uses faster links.
  - (c) bandwidth is not reserved when it is not needed.
  - (d) circuit switching uses slow modems.
18. The ABP protocol is used with packets of size  $n$ . The transmission link has a BER of  $p$ . (Assume the acknowledgments are received error free.) What is the average number of packet transmissions per correctly received packet?
19. Write a simulation of ABP with a loss probability of  $p$ . Plot the throughput as a function of  $p$  for a few values of the ratio of the propagation time over the transmission time.
20. Consider a 10,000-km round-trip route with a transmission rate of 100 Mbps. Suppose a propagation time of  $5 \mu\text{s}/\text{km}$ . Consider a packet size of 1,000 bits. How many packets are needed to fill up the links along the route? What is the minimum window size in the Go Back N protocol to achieve 100% efficiency?
21. Consider a transmission of data from S to R below. The packets have a fixed size equal to 1 KB. Each acknowledgment is in a packet with 40 bytes (consisting of an IP header and a TCP header). The packets are transmitted reliably, and acknowledgment packets are transmitted correctly with probability 0.95 each. Host R sends back an ACK with the next packet

number it expects. Host S has a timeout value equal to a “round-trip time” and implements Go Back N with a window size computed to “fill up the pipe” exactly in the absence of errors. The propagation time is 1ms in each direction. Neglect processing times.



- (a) What is the window size?
- (b) Calculate the average throughput (in packets per second) of the connection from S to R.
  
- 22. Consider the Go Back N protocol. Suppose that the packet error probability is  $p$ . (Errors in different packets are independent.) How would you calculate the efficiency, assuming that  $N$  is chosen so that the pipe is just full? How will efficiency change as  $N$  increases?
  
- 23. See the figure on the next page. Node A is sending a very large (infinite?) amount of data to Node B. The link from A to B has bit errors with probability  $p$  in  $(0, 1)$ , and the link from B to A has bit errors with probability  $q$  in  $(0, 1)$ ; these bit errors are mutually independent. An error control protocol similar to Selective Repeat is being used. This protocol is equivalent to Selective Repeat with an infinite window size. Node A sends *data packets* with  $d$  data bytes and  $h$  header bytes ( $d + h$  bytes total). Node B sends *acknowledgment packets* of  $a$  bytes.
  - (a) What is the probability that a data packet is received correctly at node B?
  - (b) What is the probability that a data packet is received correctly by node B and its subsequent acknowledgment is received correctly by node A?
  - (c) What is the expected number of times a data packet will be transmitted by node A before it is successfully acknowledged?
  - (d) Let's define efficiency to be the fraction of time node A transmits useful data (i.e., not corrupted on the way to node B, nor a duplicate copy previously delivered to node B). Headers are not considered useful data. What is the efficiency of this protocol over this link?
  - (e) Does this analysis apply to the standard Selective Repeat Protocol? If not, why?



24. Consider two transmission links. One has a Bit Error Rate (BER) of  $10^{-8}$ ; the other has BER of  $10^{-4}$ . The packets to be transmitted are 1,000 bits long. (We will ignore the acknowledgments.) A 32-bit CRC is added to the packets to check for transmission errors. The ARQ protocol is the Alternating Bit Protocol (ABP) (i.e., the link bandwidth  $\times$  delay product is small compared to the packet size). We will consider two different Forward Error Correction (FEC) codes. Both are all 31-bit BCH block codes; one uses 5 parity check bits to correct up to one error, the other uses 10 parity check bits to correct up to two errors. (The first code leaves 26 bits for user data and CRC, the second leaves 21 bits.)

Answer each of the following questions for the ARQ only case, the FEC1 plus ARQ case, and the FEC2 plus ARQ case.

- (a) How many bits are transmitted per packet?
  - (b) What is the probability that a packet is received correctly if the first link is used?
  - (c) What is the probability that a packet is received correctly if the second link is used?
25. Consider the retransmission protocol described in Figure 2.15. Suppose that the acknowledgment following a correct transmission arrives after a random delay  $d$ . (The randomness is due to random queuing delay in the network.) Let  $F$  be the cumulative probability distribution of  $d$ , namely,  $F(t) = \text{Prob}\{d < t\}$ . Thus the probability of receiving an acknowledgment before the timeout is  $F(T)$ , and the probability of not receiving it before the timeout is  $1 - F(T)$ .

- (a) Show that the expected number of timeouts that the same packet is sent is

$$N(T) = \sum_{n=1}^{\infty} nF(T)[1 - F(T)]^{n-1}.$$

Show that  $N(T)$  decreases as  $T$  increases.

- (b) Show that the expected time  $\tau(T)$  before an acknowledgment is received before a timeout is between  $T[N(T) - 1]$  and  $TN(T)$ . What is the exact value of  $\tau(T)$ ?
- (c) The timeout  $T$  is a design parameter. How would you choose it?

# Packet-Switched Networks

We saw in Chapter 2 that more sophisticated services demanded by user applications are built from basic services in a layered architecture. We also discussed the ODN architecture for communication networks. In this chapter we study the seven-layer Open Systems Interconnection (OSI) model for the logical layering of functions in data networks. The OSI model can be regarded as more detailed specifications of the ODN model, although they were developed long before the ODN model.

We then discuss the important implementations of the OSI model, beginning with the major local and metropolitan area network implementations of the *data link layer*, from Ethernet and token ring to FDDI and DQDB (Distributed Queue Dual Bus), Frame Relay, and Switched Multimegabit Data Service (SMDS). Other important implementations of the data link layer are presented in Chapters 5 and 6. We explain the Internet and the TCP/IP networks in Chapter 4.

By the end of this chapter, you will be able to estimate the limitations of each implementation in terms of speed, delay, and versatility and to judge which implementation best meets an organization's needs. You will also understand the advances that are likely to occur in the near future and what they offer.

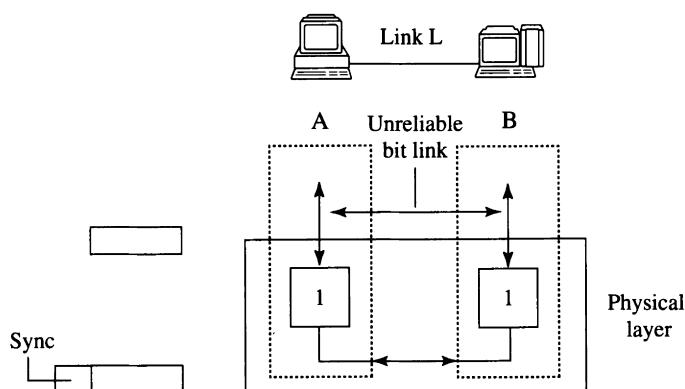
Section 3.1 describes the OSI model. Section 3.2 is devoted to Ethernet and section 3.3 to the token ring network; sections 3.4 through 3.7 describe the FDDI, DQDB, Frame Relay, and SMDS networks, respectively. The reader may skip particular sections without loss in continuity.

### 3.1 OSI AND IP MODELS

In this section we explain the Open Systems Interconnection (OSI) reference model. The OSI is a seven-layer decomposition of network functions published by the ISO. We explain the main functions performed by these layers. (See Figure 3.9 for a summary.) Many networks do not strictly follow the OSI model. In some cases, the networks were developed before the OSI model was published. However, despite these differences, the OSI model helps in understanding the design of packet-switched network architectures. The OSI model is used to specify standards.

#### 3.1.1 Layer 1: Physical Layer

The bottommost layer is the *physical layer*. It implements an unreliable bit link. A link consists of a transmitter, a receiver, and a medium over which signals are propagated. These signals are modulated electromagnetic waves that propagate either guided (by a copper cable, wire pair, or an optical fiber) or unguided in free space (as in radio). The transmitter converts the bits into signals, and the physical layer in the receiver converts the signals back into bits. The receiver must be synchronized to be able to recover the successive bits. To assist the synchronization, the transmitter inserts a specific bit pattern, called a *preamble*, at the beginning of the packet, indicated by sync in Figure 3.1. The



**FIGURE**

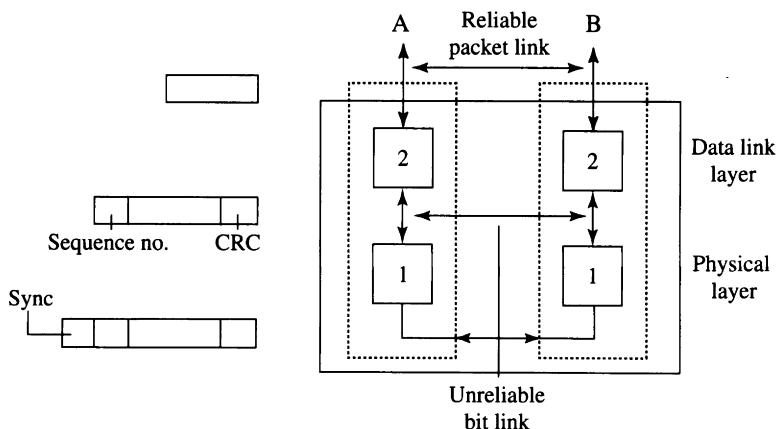
The physical layer transmits bits by converting them into electrical or optical signals. In many implementations, the physical layer uses synchronization bits (sync) to synchronize the receiver.

bit link is unreliable because synchronization errors and noise in the link can corrupt a packet.

Physical layer standards specify the modulation scheme (the relation between the bits and the electromagnetic signal), and the characteristics of the interface between the transmitter and receiver and the medium. A link's characteristics impose a limit on how fast it can transmit data. Generally, wireless links are slower than copper links, and copper links are slower than optical links.

### 3.1.2 Layer 2: Data Link Layer

Figure 3.2 illustrates the *data link layer* for a point-to-point link between two computers. As we explained in our discussion of error detection (see section 2.6.3), the transmitter and receiver can execute a specific protocol to retransmit corrupted packets. In the figure, the protocol entities are represented by the boxes labeled 2. These entities are programs that are usually executed by dedicated electronic circuits (in the NIC or network interface card) because of the high speed.



**FIGURE**  
3.2

The data link layer supervises the transmission of packets by the physical layer. In a typical implementation, the data link layer adds a sequence number and error-detection bits (CRC). As we explain in the text, networks with reliable links control the errors from source to destination instead of controlling them on every link.

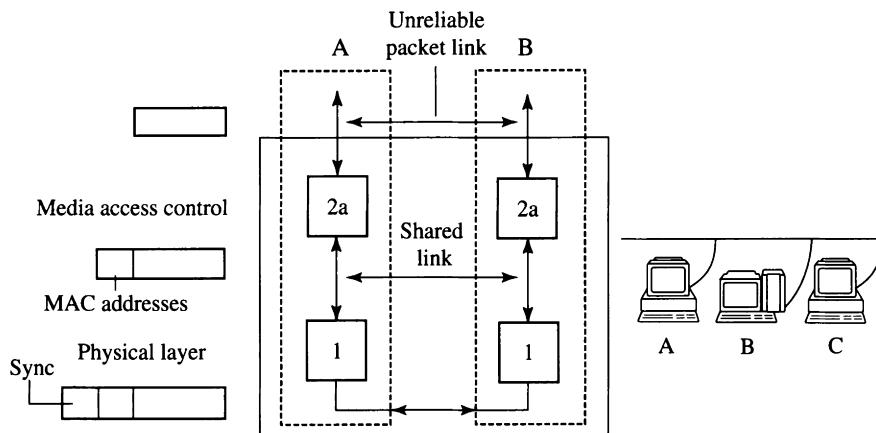
The transmitter appends error-detection bits and may number the packets. The figure shows the fields that are appended to the packet that the data link layer transmits; first, the data link layer in the transmitter adds the error-detection bits (CRC) and may add a sequence number to the packet. Then the physical layer adds a synchronization preamble (sync). In the receiver, the physical layer strips the preamble and gives the rest of the packet to the data link layer, which uses the error-detection bits to verify that the packet is correct. If the data link layer arranges for retransmissions of erroneous packets, it uses the packet sequence numbers to determine which packet should be retransmitted. If the data link layer only drops incorrect packets, then it does not insert a sequence number. The data link layer then strips the error-detection bits and the sequence number (if present).

The errors can be controlled end-to-end instead of hop-by-hop (at each link). End-to-end control is preferable to link-level control when the network uses links with a small bit error rate. Indeed, in such a situation, most packets reach their destination without errors, and it is wasteful to verify them at every link. End-to-end error control is implemented at the transport layer (layer 4) with the same mechanism used by the data link layer. Link-level error control is preferable in links with a high bit error rate, as in wireless radio or satellite links. In these links error-correction bits may be used because retransmission of corrupted packets may incur large delays.

The fields that the protocol entities add to the packet contain control information that the protocols in the different layers use to monitor the transmissions. The appending of control fields to a packet is called *encapsulation*. The reverse process, stripping the control fields, is *decapsulation*. Observe that encapsulation and decapsulation may often be performed without examination of the packet, which simplifies the hardware and reduces packet processing time.

### 3.1.3 Sublayer 2a: Media Access Control

Figure 3.3 shows computers attached to a common link. These computers must regulate the access to that shared link. This function, called access control, is performed by a sublayer called the *media access control* (MAC) sublayer. Thus, the MAC sublayers in the computers follow a set of rules—a protocol—to regulate access to the shared link. Because the link is shared, the MAC must append the physical address of the destination, the specific computer to which the packet is destined. The physical address identifies uniquely a



**FIGURE**  
3.3

The figure shows three computers that share a common link. Access to such a common link is regulated by the media access control (MAC) sublayer. That sublayer adds the addresses of the source and destination to the packet before giving it to the physical layer for transmission.

device attached to a shared link. Such a physical address is not needed for a point-to-point link.

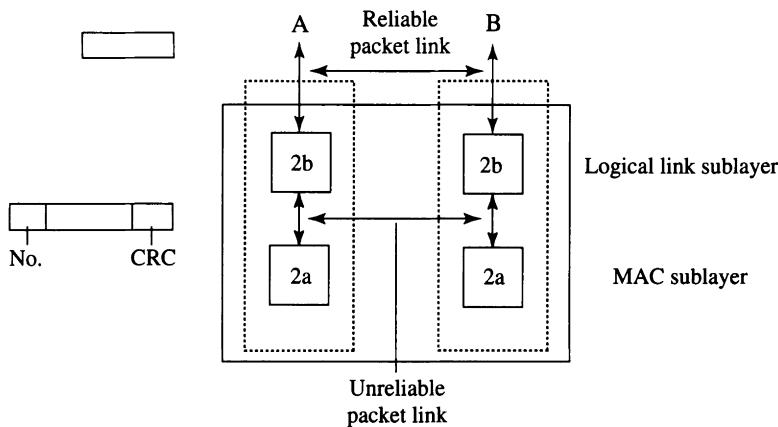
The common link may be a copper cable to which the computers are attached (as in Ethernet and CATV), or a radio broadcast channel to which all the computer radio receivers are tuned.

The MAC sublayer implements the unreliable transmission of packets between computers attached to the common link. MAC standards specify the packet formats, the MAC addressing scheme, and the MAC protocol.

### 3.1.4 Sublayer 2b: Logical Link Control

Figure 3.4 shows the *logical link control* (LLC) sublayer. It uses the unreliable transmission of packets implemented by the MAC sublayer to implement either only error detection or reliable packet transmission between computers attached to a shared link. The functions of the LLC are the same as those the data link layer executes for a point-to-point link.

The MAC and LLC together constitute the data link layer for multiple access links: they use the unreliable bit link of the physical layer to implement a packet-transmission service with error detection or a reliable packet-transmission service between computers attached to a common link.

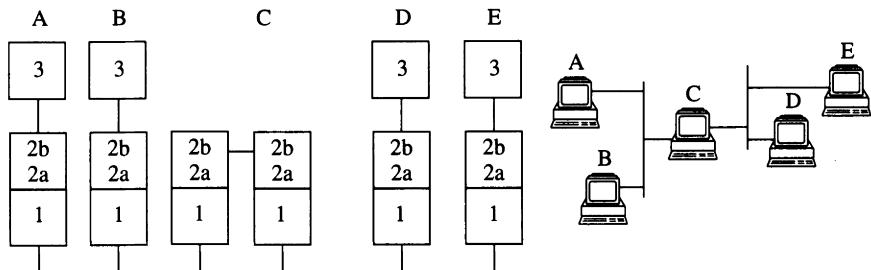


**FIGURE**  
3.4

The logical link control (LLC) adds error detection to the transmissions that the MAC sublayer implements. (The LLC can also use a retransmission mechanism to provide reliable packet transmissions, if desired.)

Figure 3.5 shows a *bridge* (computer C) between two Ethernet networks. Computers A, B, and C are attached to one Ethernet. The right part of the figure shows these three computers attached to the same link. The MAC sublayer in the three computers implements unreliable packet transmissions. The LLC detects the errors in the transmissions. The situation is similar for computers C, D, and E attached to the other Ethernet.

Consider a packet sent by computer A and destined for computer E. Computer C must store that packet and retransmit it on the second Ethernet. If the packet sent by A is destined for computer B, C must not retransmit it. The



**FIGURE**  
3.5

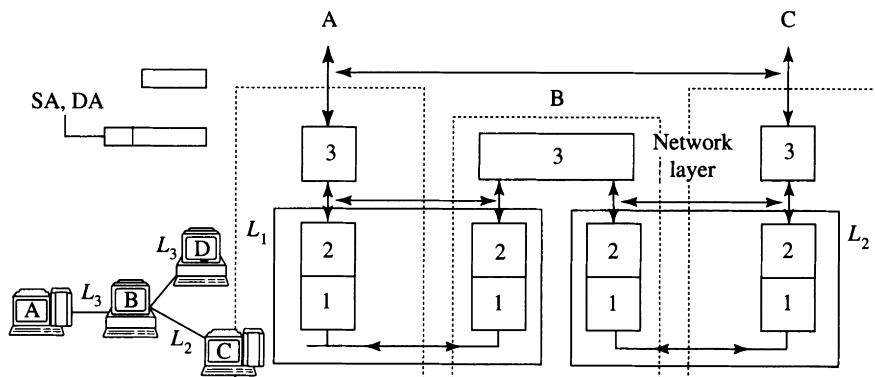
Computer C in this figure is a *bridge*. It connects two local area networks by copying packets from one to the other.

decision of C is limited to whether to retransmit the packet or not. A bridge is a computer equipped with the hardware and software to perform such decisions and capable of retransmitting packets at a high rate. The bridge allows the computers on the two Ethernets to function as if they are on the same Ethernet (see section 3.2.3).

### 3.1.5 Layer 3: Network Layer

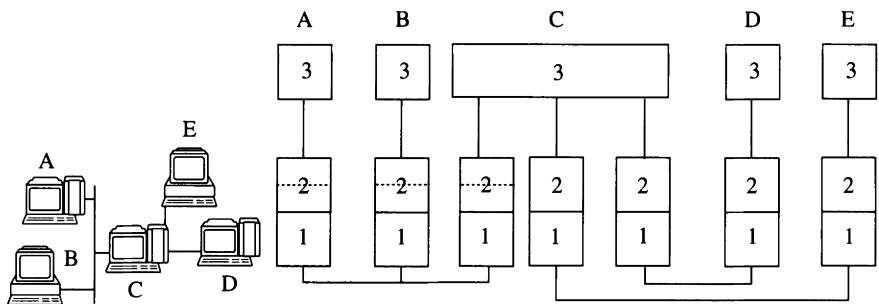
The data link layer implements a packet link between computers attached to a common link. As we explained in our discussion of store-and-forward packet switching (see section 2.6.2), when computers are connected by a collection of point-to-point links, they must figure out where to send the packets that they receive: whether to send them out over another link and, if so, which one. (See Figure 3.6.) This function—finding the path the packets must follow—is called *routing*. Routing is one of the main functions of the *network layer*. The network layer appends unique network addresses of the source and destination computers. An important addressing scheme in packet-switched networks is that used by the Internet. Circuit-switched networks, like the telephone network, use different addressing schemes.

Thus, the network layer uses the transmission over point-to-point links provided by the data link layer to transmit packets between any two computers attached in a network.



**FIGURE**  
3.6

The network layer delivers packets between any two computers attached to the same network. That layer implements store-and-forward transmissions along successive links from the source to the destination.



**3.7  
FIGURE**

Computer C in this figure is a *router*. It is designed to relay packets at a high rate to the proper link and with a low delay.

Figure 3.7 shows a *router* attached to several links. When the router receives a packet, it must decide on the basis of the network addresses along which link it should retransmit the packet. This routing function is implemented by the network layer.

Observe that the link between, say, C and D in Figure 3.7 may carry packets between A and E and between B and E. These packets are statistically multiplexed by the router C (see section 2.6.1). The network addresses of the packets permit demultiplexing. Network layer standards specify packet formats, addressing schemes, and routing protocols.

### 3.1.6 Layer 4: Transport Layer

The *transport layer* delivers *messages* between *transport service access points* (TSAPs or *ports*) in different computers. Several processes running on a computer may be exchanging messages with processes running on other computers. The TSAPs appended to the messages differentiate those information streams. The transport layer may multiplex several low-rate transmissions with different TSAPs onto one virtual circuit or divide a high-rate connection into parallel virtual circuits.

Some frequently used applications such as e-mail and file transfers are allocated fixed TSAPs (also called *well-known ports*). To connect to a process with unknown TSAP, a remote process first connects to a process server attached to a fixed TSAP. The server then indicates the TSAP of the desired process.

The transport layer delivery of messages is either *connection-oriented* or *connectionless*. A connection-oriented transport layer delivers error-free messages in the correct order. Such a transport layer provides the following services:

CONNECT, DATA, EXP\_DATA, and DISCONNECT. CONNECT sets up a connection between TSAPs. DATA delivers a sequence of messages in the correct order and without errors. EXP\_DATA delivers urgent messages by making them jump ahead of the nonurgent messages in the two end nodes. DISCONNECT releases the connection. A connectionless transport layer delivers messages one by one, possibly with errors, and with no guarantee on the order of the messages. The service of a connectionless transport layer is UNIT\_DATA, the connectionless delivery of a single message.

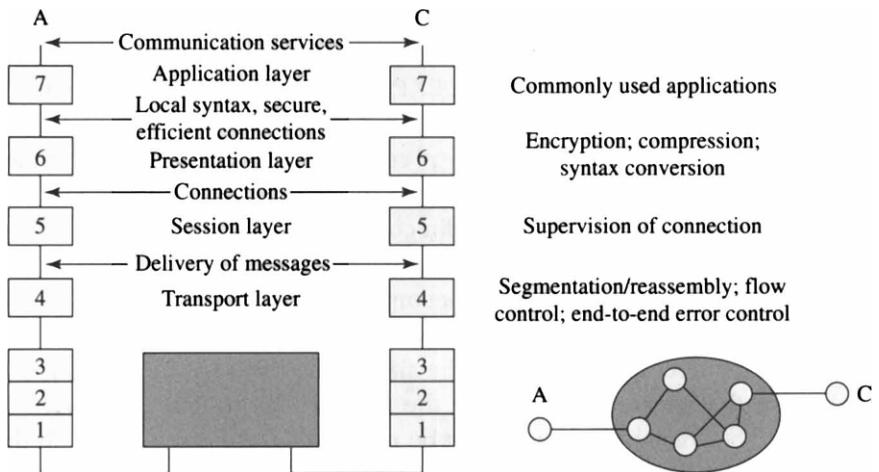
The transport layer decomposes messages into packets and combines packets into messages, possibly after resequencing them. Such packetization may be necessary because the size of the message may be larger than the size of the packets accepted by the network layer. Resequencing may be necessary because packets may not be received in the order in which they are transmitted. To perform this packetization function, the transport layer numbers the packets belonging to the same message. The layer also controls the flow of packets to prevent the source from sending packets faster than the destination can handle them. Moreover, the transport layer requests retransmissions of corrupted packets (see Figure 3.8).

To be able to combine packets into messages, the transport layer numbers packets. When the source computer fails, it may lose track of the sequence number it had reached. Resuming the numbering arbitrarily might lead to delayed packets in the network having the same sequence numbers as the packets being transmitted; this would confuse the destination. One solution to this problem is to attach a “time to live,”  $L$ , to each packet and to decrement that time to live every time the packet goes through a network node. Packets with a zero time to live are discarded. Thus,  $L$  is the maximum number of hops for each packet, and it corresponds to a maximum lifetime of  $T$  seconds inside the network, because a packet will remain in each node for some bounded time. When a source computer recovers from a crash, it can wait for  $T$  seconds to make sure that all the delayed packets to the destination were discarded, thus avoiding any numbering ambiguity. The transport layer protocols in the two nodes agree on an initial sequence number by using a three-way handshake.

Figure 3.8 summarizes the functions of layers 4 to 7.

### 3.1.7 Layer 5: Session Layer

The *session layer* supervises the dialog between two computers. It can set up a connection prior to an exchange of information between the machines. The session layer can partition a transfer of a large number of messages by inserting



3.8

FIGURE

This figure summarizes the functions and services of layers 4 to 7.

*synchronization points*. These synchronization points are specific packets that divide the sequence of messages into groups. In case of computer malfunction, the transmission can restart from the last synchronization point.

### 3.1.8 Layer 6: Presentation Layer

Application programs and computer devices use different conventions to represent information by binary numbers. For instance, some computers represent 16-bit words by placing the most significant byte before the least significant byte, whereas other computers use the opposite convention. As another example, different terminals use different control characters to specify backspace, line feed, and carriage return. Also, different application programs may follow different rules to encode data structures such as matrices and complex numbers. Computer scientists refer to a set of rules for representing information as a *syntax*. Thus, different computers use different syntaxes. The syntax used by a computer is its *local syntax*.

Suppose that computers using  $N$  different local syntaxes want to communicate. One possible method is to have every computer perform the  $N - 1$  syntax conversions needed to communicate with the other computers. Another method is to adopt a common transfer syntax and have every computer perform the conversion between its local syntax and the transfer syntax. The second

method is obviously more convenient because it does not require a computer to be aware of all the possible other local syntaxes. Communication networks use the second method, and the presentation layer is responsible for the conversion between local syntax and transfer syntax. As shown in Figure 3.8, one service provided by the presentation layer is the exchange of information between computers, each using its own local syntax.

An additional task of the presentation layer is to encrypt transmissions that must be secure. In abstract terms, encryption is a one-to-one transformation of a message into an encrypted version.

Another important task of the presentation layer is data compression. Such data compression eliminates some of the redundancy in the information to be transmitted, thereby reducing the number of bits to be transferred.

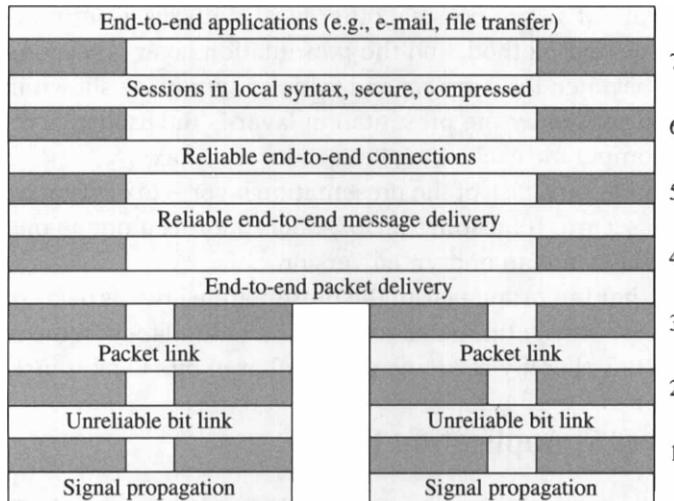
### 3.1.9 Layer 7: Application Layer

The *application layer* provides frequently needed communication services such as file transfer, terminal emulation, remote login, directory service, and remote job execution. These services are used by the user applications. For instance, an e-mail program uses a file transfer service that delivers a file to a list of addresses and informs the sender if some addresses cannot be reached. User applications, such as e-mail or WWW, are run on top of the application layer.

### 3.1.10 Summary

The OSI model is a detailed architecture that specifies how complex services such as file transfer and remote login are to be constructed out of the most basic services of unreliable bit transfer in a seven-layer hierarchy. Each layer adds functionality to the services of the layer immediately below it. The OSI model is summarized in Figure 3.9.

We can use the OSI reference model to add more detail to the simple diagram of the network elements in Figure 2.1. In the figure, elements labeled "Switch" are connected to other switches by links. From the OSI reference model we now know that a computer at a switch or at a user can connect to another computer at different layers. For example, if the switch element is a router like computer C in Figure 3.7, it connects with neighboring computers at the network layer. In this case, the switch only need implement layers 1, 2 and 3. If the switch element is a bridge like computer C in Figure 3.5, it connects with its neighbors at the data link layer, so that it only need implement layers 1 and 2. The user host computer runs applications and so it must implement all seven layers.



**3.9  
FIGURE**

The figure shows the services implemented by the seven layers of the OSI reference model.

We can place the OSI model in the context of the Open Data Network model of section 2.8 by identifying layers 1 and 2 with bit ways, layers 3 and 4 with bearer services, layers 4, 5, and 6 with middleware, and layer 7 with applications.

Most network implementations do not follow the OSI model, but rather the four-layer IP model based on the Internet protocol layers shown in Figure 4.2.

## **3.2      ETHERNET (IEEE 802.3)**

Ethernet is the most popular local area network or LAN technology. More than 100 million Ethernet nodes have been deployed, and more than 50 million Ethernet network interface cards are sold each year. The installed base and market share of Ethernet networks dwarfs those of other types of LANs. The large installed Ethernet base has also spurred a series of advances, such as fast and gigabit Ethernet and Ethernet switches, which are backward-compatible with earlier versions of Ethernet.

Ethernet is specified by the IEEE 802.3 standards, published in 1985. An Ethernet LAN typically uses twisted pair wires or fiber optic cable. 10BASE-T

(Ethernet over twisted pairs) transmits at 10 Mbps, and 100BASE-T or Fast Ethernet transmits at 100 Mbps. Gigabit Ethernet, with a speed of 1,000 Mbps (1 gigabit or 1 billion bps), usually over optical fiber, is used for enterprise-wide backbones.

Ethernets are inexpensive and provide a relatively high throughput and low delays that can support many applications. Most importantly, Ethernet provides inexpensive, relatively high-speed network access to individual users. (The token ring network, in which users share some of the transmission costs, also provides inexpensive access. The CATV network, described in Chapter 5, provides shared access over longer distances.)

### 3.2.1 Physical Layer

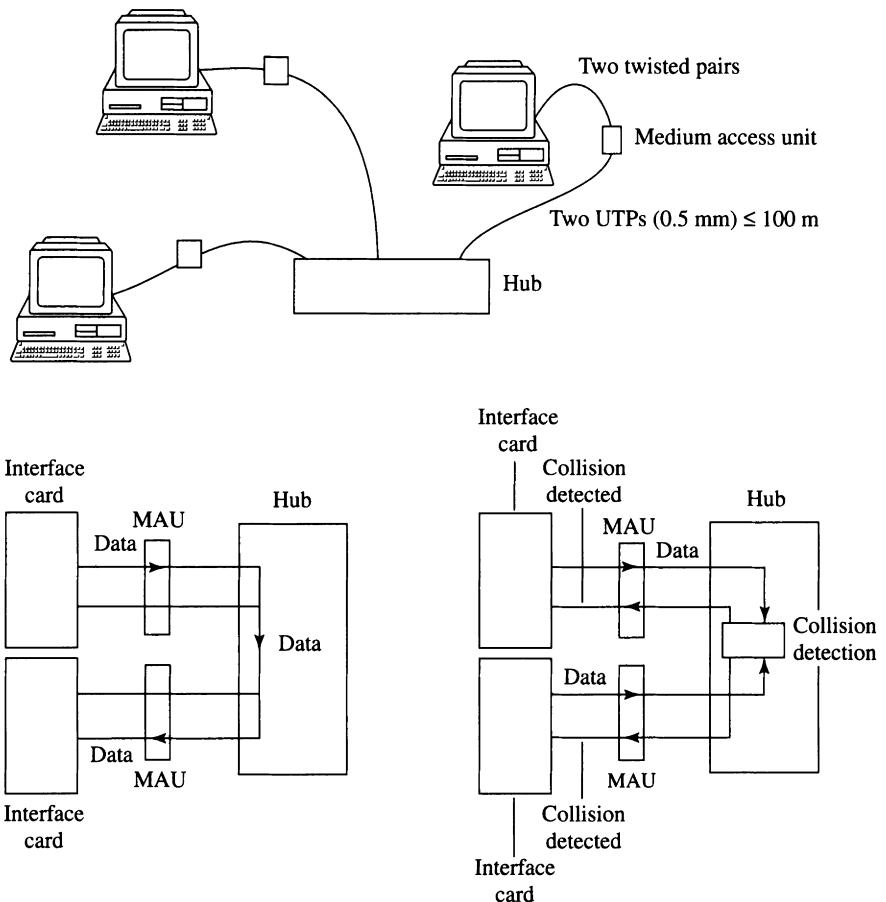
The physical layer of IEEE 802.3 networks specifies the electrical and mechanical characteristics of the wiring and of the encoding of the bits. We describe the physical layers of the 10BASE<sub>x</sub>, 100BASE<sub>x</sub>, and the 1000BASE<sub>x</sub> networks, with transmission speeds of 10, 100, and 1,000 Mbps, respectively.

#### 10BASE-T

The 10BASE-T wiring, shown in Figure 3.10, may use the same category 3 unshielded twisted pairs (UTPs or UTP-3s) that telephone companies use to wire buildings for telephone service. Consequently, if an office building or residence has enough spare UTPs, these can be used to install a 10BASE-T network. (This explains the popularity of this form of Ethernet. Economies of scale have led to a steady reduction in the cost of Ethernet chips and boards. Today, the cost of wiring is likely to be more than the cost of the Ethernet hardware.)

In a 10BASE-T network, each computer is equipped with a network interface card that is attached to a medium access unit (MAU) with two twisted pairs. Two unshielded twisted pairs attach an MAU to a 10BASE-T hub. Hubs can be attached together with two UTPs to build larger networks. A hub is a repeater: it transmits a packet received on one port to all the other ports.

The physical layer encodes the bits as electrical signals using the Manchester encoding. In this encoding, the bits are represented by a two-value signal that makes a transition during every bit transmission. Specifically, a bit 0 is encoded as the low value for half of the bit transmission interval followed by the high value for the second half of the interval. A bit 1 is encoded as the high value for the first half of the interval followed by the low value for the second half of the interval. At 10 Mbps, the bit transmission interval lasts 0.1  $\mu$ s.



**3.10**  
**FIGURE**

Physical layout of a 10BASE-T network. This network uses unshielded twisted pairs and can be wired with spare telephone pairs already in place in the building. The transmission rate is 10 Mbps. One can attach hubs together to build larger networks.

The receiver uses the transitions to synchronize to the received signal. Such a code is said to be *self-synchronizing*. The actual receiver hardware consists of a phase-locked loop that adjusts its phase to match the transitions of the received signal.

### 100BASE-T

The 100BASE-T transceivers operate at 100 Mbps over two pairs of UTP-3s, the same as for 10BASE-T. (Thus 10BASE-T Ethernet users can upgrade to 100BASE-T

without changing their cabling infrastructure.) The maximum distance between two computers is 100 m. Physical layer standards are also defined for other cabling arrangements, including the following: 100BASE-FX is for transmission over optical fiber, 100BASE-TX is for full-duplex transmission over two pairs of UTP-5 (category 5) cables or two pairs of shielded twisted pairs (STP), 100BASE-T4 is for four UTP-3 cables. The modulation schemes for all these cabling arrangements are different.

Similar to 10BASE-T, the 100BASE-X family (except T4) supports simultaneous or full-duplex transmission of 100 Mbps data streams on one link segment in each direction.

The standards also specify an auto-negotiation protocol that promotes flexible products, such as dual-speed 10/100 Mbps network interface cards.

### **1000BASEx**

The physical layer standard for gigabit Ethernet was approved in June 1998. It also permits several arrangements: 1000BASE-LX is for transmission over single- or multimode fiber, 1000BASE-SX is for multimode fiber, and 1000BASE-CS is for shielded copper cables. A future standard, 1000BASE-T, will consider transmission over UTP.

### **Wireless Ethernet**

A number of commercial products are available to set up wireless Ethernet networks. In such a wireless Ethernet network, all the stations share a radio channel. The physical layer standards specify the frequency spectrum that the radio channel can use, and the modulation scheme. There are many varieties of wireless LANs. Some LAN products diffuse infrared light signals instead of radio waves.

### **10BASE5**

Prior to the introduction of 10Base-T and 100Base-T, Ethernet networks used the 10BASE5 version. We briefly describe this version for historical interest and because such networks are still around. The wiring of a 10BASE5 consists of segments. Each segment is a length of up to 500 m of coaxial cable with a diameter of 10 mm and a characteristic impedance of 50 ohms. Segments are connected by repeaters that can be up to 1,000 m apart. No two computers on the network can be more than 2,500 m apart. The transmission rate is 10 Mbps; the physical layer uses the Manchester encoding.

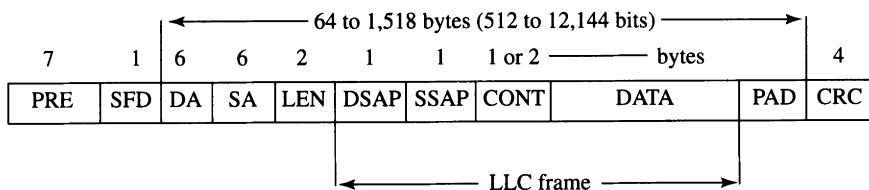
### 3.2.2 MAC

The media access control sublayer of Ethernet specifies the MAC addresses of network interfaces, the frame format, and the MAC protocol for sharing the cable.

The MAC address of an Ethernet interface card is a 48-bit string set by the manufacturer that is unique to that card. The first 24 bits of the address are the “organizational unique identifier” that the IEEE assigned to the manufacturers of the Ethernet interface. Each manufacturer then makes sure that its interfaces have different lower 24-bit identifiers.

The frame format of Ethernet packets is shown in Figure 3.11. The preamble (PRE) synchronizes the receiver. The start-of-frame delimiter (SFD) indicates the start of the frame. The destination (DA) and source (SA) MAC addresses are 48-bit-long strings unique to each interface card. The length indicator (LEN) eliminates the need for an end-of-frame delimiter and permits the use of a padding field (PAD) to make sure that the frames have at least 64 bytes. The cyclic redundancy check (CRC) enables the receiver to detect most transmission errors, as we explained in section 2.6.3. The logical link control (LLC) frame specifies the destination (DSAP) and source (SSAP) service access points. A slightly different standard (called the *DEC-Intel-Xerox* or *DIX Ethernet*) replaces the length field by a 2-byte type field.

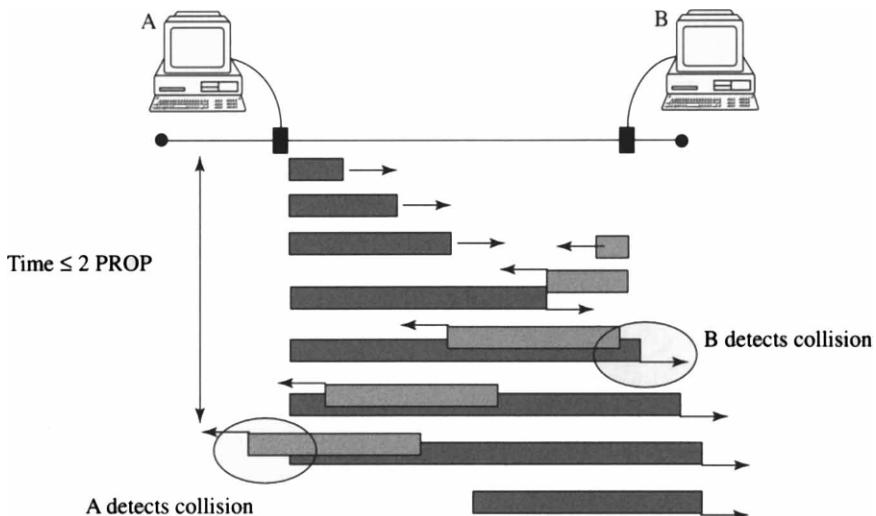
The Ethernet MAC protocol is CSMA/CD, Carrier Sense Multiple Access with Collision Detection. When using CSMA/CD, a node that has a packet to transmit waits until the channel is silent before transmitting. Also, the node aborts the transmission as soon as it realizes that another node is transmitting. After aborting a transmission, the node waits for a random time and then repeats these steps. (Note that the CSMA/CD protocol is disabled in full-duplex operation. Such operation is possible only between devices equipped with buffers.)



3.11

FIGURE

Format of Ethernet packets. The various fields that make up the frame are explained in the text.



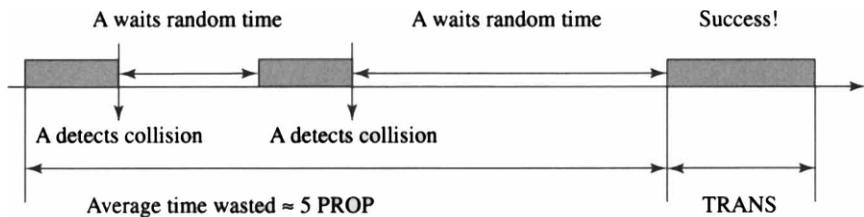
3.12

**FIGURE**

The maximum time until a node detects a collision is twice the propagation time of a signal between the nodes that are farthest apart.

Figure 3.12 shows two nodes A and B that start transmitting. Node B starts transmitting shortly before the signal sent by node A reaches it. Node A detects that node B is also transmitting after about one round-trip propagation time of a signal between nodes A and B. The protocol requires that A detect the collision before it has stopped transmitting its packet; thus the transmission time of the smallest packet must be larger than one round-trip propagation time. This requirement limits the maximum distance between two computers on the network. The smallest packet is 64 bytes. This 64-byte value is derived from the original 2500-m maximum distance between Ethernet interfaces plus the transit time across up to four repeaters plus the time the electronics takes to notice the collision. The 64 bytes correspond to  $51.2 \mu\text{s}$ , which is larger than the round-trip time across 2500 m (about  $18 \mu\text{s}$ ) plus the delays across repeaters and to detect the collision.

The time taken to transmit a packet that collides is wasted because the two nodes must abort their transmission and restart after some random delay. This wasted time is proportional to the propagation time between the nodes. Figure 3.13 illustrates the sequence of events that takes place when a node transmits a packet. The figure shows that the node starts transmitting, then detects that another node is also transmitting so that it must abort its transmission, and it then waits for some random time before trying again. Eventually, the node succeeds in transmitting its packet.



**3.13  
FIGURE**

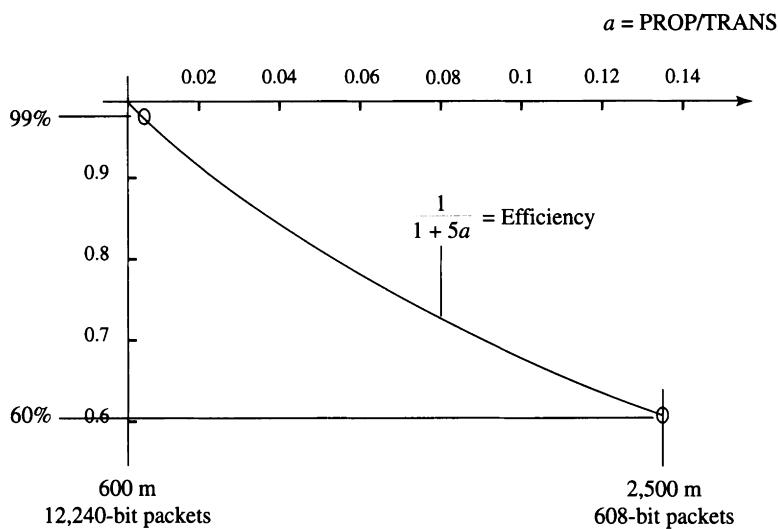
Sequence of events on an Ethernet network. The transmitters waste five propagation times per successful transmission, on the average.

It can be shown that when many nodes attempt to transmit, they waste, on the average, an amount of time approximately equal to  $5 \times \text{PROP}$  per successful transmission. Here, PROP designates the propagation time of a signal from one end of the cable to the other. Accordingly, the fraction of time that the nodes use the transmission channel to transmit packets successfully is approximately equal to  $1/(1 + 5\alpha)$  where  $\alpha$  denotes the ratio of a propagation time (PROP) to a packet transmission time (TRANS). This fraction of useful time when many nodes want to transmit is called the *efficiency* of the MAC.

For instance, if  $\alpha$  is very small, then the nodes learn very quickly about conflicting transmissions. Consequently, the nodes waste very little time because of such conflicts, and the efficiency of the MAC is close to unity. At the opposite extreme, if the packet transmission times are comparable to a propagation time, then a node learns of simultaneous transmissions only after a long time (when the signal from another node reaches it), so that a significant fraction of time is wasted by transmissions that are aborted, and the efficiency is small.

Figure 3.14 shows two representative values of the efficiency of Ethernet. The first value corresponds to a relatively short Ethernet (600 m) with packets that have the maximum length admissible by the Ethernet standards. In that case, the calculations show the efficiency to be about 99%. The other case corresponds to an Ethernet with the maximum admissible separation between nodes (2,500 m) and with the smallest possible Ethernet packets. In this least-efficient case, the efficiency is about 60%. In a typical situation, the efficiency can be expected to be about 80%. This means that out of the raw bit rate of 10 Mbps, 80%, or 8 Mbps, are used to transmit successful packets.

In a 10BASE5 Ethernet, computers share the same medium so transmissions are broadcast, collision is possible and detected by all computers, and the CMSA/CD protocol permits recovery from the resulting packet errors. When computers are connected over UTPs to a common hub as in Figure 3.10, the



3.14

FIGURE

Two representative values of the efficiency of Ethernet. Typically, the efficiency is about 80%.

medium is not shared. However, the hub, which repeats each packet on every port, converts it into a broadcast medium. When a hub detects a collision (more than one port has an incoming signal), it sends a jamming signal to all ports, to emulate collision detection.

When several hubs are interconnected to build a larger network, a packet is repeated on the ports of all those hubs. The computers attached to those hubs form a single *broadcast domain* and share the total bit rate. Bridges and switches, by contrast, selectively repeat packets received at a port, thereby segmenting a broadcast domain, so that packets on different segments can be simultaneously transmitted. The aggregate bit rate is thereby increased.

In a switched Ethernet, a switch replaces the hub and collisions are not possible. However, for backward compatibility, the same frame structure is used.

### 3.2.3 LLC

The IEEE 802.2 logical link control standard is used for 802.3, 802.5, and other networks. The LLC sublayer provides connection-oriented or connectionless (acknowledged or not) services. The LLC can also multiplex different transmissions that are differentiated by the service access point field (see Figure 3.11).

Finally, the LLC implements the *transparent routing* of packets between Ethernets attached together with bridges.

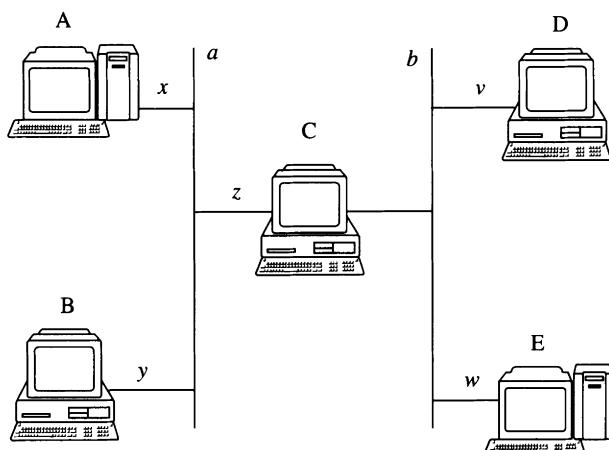
When it provides a connection-oriented or an acknowledged connectionless service, the LLC uses the CRC field to detect errors. To implement a connection-oriented service, the LLC uses the Go Back N protocol (see section 2.6.3) to arrange for the transmitter to retransmit packets that do not arrive error free.

### 3.2.4 LAN Interconnection

LANs can be interconnected by hubs, bridges, and switches. These are layer 2 devices. Since hubs are repeaters, attaching two LANs to the same hub converts them to a single LAN segment so that computers on both LANs share the same bit rate. Bridges and switches are used to connect LANs but instead of repeating packets, they forward them selectively.

#### Bridges

We use Figure 3.15 to explain how a packet with a given MAC destination address finds its way in a network of Ethernets connected by bridges. The bridges



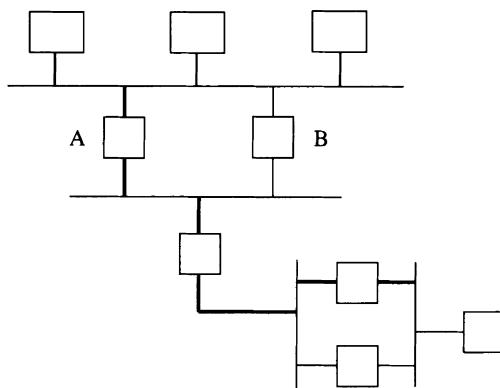
3.15

**FIGURE**

Ethernet attached by bridges. Transparent routing enables a computer to transmit a packet to another computer on one of these Ethernets as if it were on the same Ethernet.

use a simple procedure called *transparent routing*. Transparent routing requires the bridges (such as node C in the figure) to maintain tables of MAC addresses of the nodes on their Ethernets. To maintain such a table, a bridge reads the source addresses of the packets broadcast on the Ethernets. Say that a packet on an Ethernet is *local* if it is destined for another node of the same Ethernet. By maintaining address tables, a bridge learns which packets are local. When bridge C receives a packet that it does not think is local, bridge C retransmits the packet on the other Ethernet. If the packet was in fact local, no damage is done since all the nodes on the other Ethernet will disregard the packet.

Transparent routing is a very convenient procedure. However, it can make packets loop forever between bridges unless some care is taken to avoid that behavior. Consider Figure 3.16. Assume that the top-left node in the figure sends a packet destined to the rightmost node. The packet is seen by two bridges, say A and B. Bridge A retransmits the packet on the second Ethernet because the packet is not local. When bridge B sees that packet on the second Ethernet, it retransmits it on the top Ethernet because the packet is not local to the second Ethernet. Thus bridge A gets the packet a second time, and the bridges A and B repeat this sequence of retransmissions indefinitely. To prevent such an infinite loop, the bridges use *spanning tree routing*. A spanning tree in a graph is a subgraph that is a tree (i.e., loop-free) and that spans all the nodes (but not necessarily all the bridges). If the bridges know a tree that spans all



3.16

FIGURE

One method that bridges use to avoid making multiple copies with transparent routing is spanning tree routing. Only the bridges along the spanning tree shown by thick lines copy the packet.

the network nodes, then they can avoid loops by agreeing that the packets will be retransmitted only by the bridges on the tree. A spanning tree is shown by thicker lines in Figure 3.16.

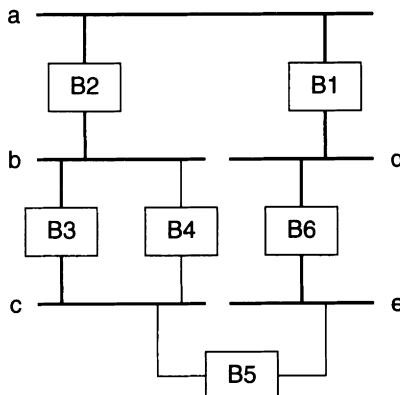
The bridges construct a spanning tree by using a distributed algorithm that finds the shortest path to destinations. The shortest path to any destination cannot contain any loop. Consequently, the set of shortest paths from all the bridges to a particular bridge must be a spanning tree.

We explain the spanning tree algorithm that IEEE802 networks use on the network of Figure 3.17. The key idea of the algorithm is that the root is elected at the same time that the spanning tree is constructed. The root is the bridge with the smallest identification number. In this network, five Ethernet LANs ( $a, \dots, e$ ) are connected with six bridges ( $B1, \dots, B6$ ). The bridges have identification numbers (e.g., their IP addresses), and in our notation, these identification numbers are increasing from  $B1$  to  $B6$ .

Each bridge sends messages of the following format:

[ID.sender|ID.presumed\_root|distance.presumed\_root]

where ID.sender is the identification number of the bridge that sends the message, ID.presumed\_root is the identification number of the bridge that the sender believes to be the root of the spanning tree, and distance.presumed\_root is the current estimate of the number of hops between the sender and the presumed root. Each bridge stores the best message that it has received so far



3.17

FIGURE

In this figure,  $a, \dots, e$  are Ethernet LANs and  $B1, \dots, B6$  are bridges.

from each port. A message  $[X|R|D]$  is better than a message  $[X'|R'|D']$  if

$$R < R', \text{ or}$$

$$R = R' \text{ and } D < D', \text{ or}$$

$$R = R' \text{ and } D = D' \text{ and } X < X'.$$

When it learns that it is not the root, a bridge stops sending messages. When it gets a better message on a port than the ones it has sent so far on that port, it stops sending messages on that port and only relays other messages after adding 1 to their distance. Eventually, only the bridge with the smallest ID (the root) sends messages and the others relay them.

Here is the sequence of messages sent by the bridges in Figure 3.17:

- ◆ Initially,  $B_i$  sends  $[B_i|B_i|0]$  on all its ports.
- ◆ When  $B_2$  gets  $[B_1|B_1|0]$ , it stops transmitting its own messages and it sends  $[B_2|B_1|1]$  on all its lower ports.
- ◆ Similarly  $B_6$  sends  $[B_6|B_1|1]$ .
- ◆ When  $B_4$  gets  $[B_3|B_3|0]$ , it stops sending and relays  $[B_4|B_3|1]$ . When  $B_4$  later gets  $[B_2|B_1|1]$ , it then relays  $[B_4|B_1|2]$ .
- ◆ When  $B_5$  gets  $[B_3|B_1|2]$  and  $[B_6|B_1|1]$ , it finds out that it is at distance 2 from the root  $B_1$ , and it then learns that none of its ports are on the shortest path to the root.

To accommodate changes in topology, the bridges keep sending these messages. If a bridge  $B_j$  stops receiving messages  $[B_i|B_1|di]$ , then after a timeout it claims again to be the root by sending the messages  $[B_j|B_j|0]$  and the algorithm restarts.

### *Ethernet Switches*

An Ethernet switch is a multiport bridge that selectively forwards packets from one LAN port to another port. The bit rate on different ports may be different. Like hubs and bridges, switches may be interconnected to form larger networks. A switch's forwarding decision is based solely on layer 2 information. Switches do not modify the received packet. (Routers, by contrast, base their forwarding decision on layer 3 or network layer information, and also modify the received packets.)

Packets destined for different ports may be simultaneously forwarded by the switch, so a switch can increase the overall bit rate many times compared

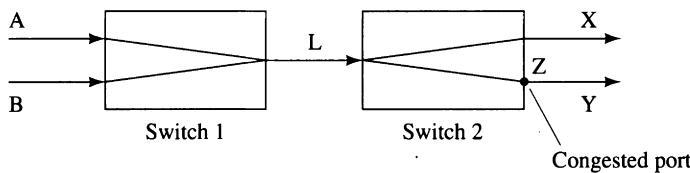
with a single shared LAN. But packets destined for the same port must be buffered by the switch. Thus a switch consists of a switching fabric, buffers, and the forwarding control mechanism. The switch fabric may be blocking or nonblocking, and the buffers may be segregated by either input port or output port or ports may share buffers. The fabric design and buffer management affect switch performance as discussed in Chapter 12.

The forwarding mechanism builds a table relating the MAC address of a computer on a LAN to the port number to which the LAN is connected. Such a table is built by associating the MAC source address of an incoming packet with the number of the incoming port. If a destination address cannot be resolved by the table, the switch, similarly to a bridge, sends the packet out on all the other ports. This can happen if the destination computer is beyond the LANs directly connected to the switch. Broadcast packets are sent to all the ports, except in the case of VLANs (see next page).

Performance depends on other features of switch design. A switch may forward a packet after it has been fully received (store-and-forward), or it may start forwarding as soon as the output port has been determined (cut-through). Although cut-through forwarding can clearly reduce latency, the switch cannot carry out a CRC check and so corrupted packets are forwarded.

Switches are deployed to improve LAN performance because they can increase the bit rate available to alleviate congestion and to match the bit rate to the traffic on different LANs. Thus, for example, in client/server networks, the clients would be connected to a switch's 10 Mbps ports and the servers would be connected to its 100 Mbps ports. However, if the traffic on switched LANs does not match port speeds, congestion can occur and degrade performance, as we discuss next.

The switch buffers can temporarily store packets contending for the same port. But if contention persists, the buffers will fill up and the switch will drop packets unless there is a flow control mechanism that sends a signal stopping the source from sending additional packets. (In the OSI model flow control is a function of the transport layer, layer 4, and not the link layer.) However, link-based flow control stops traffic on an entire link rather than stopping only the particular source responsible for the congestion. Link-based flow control can thereby interfere with an uncongested path. In the switched LAN of Figure 3.18, A is sending packets to X and B to Y. Both paths share link L. If the port at Y is congested (possibly because another source is also sending packets to Y), switch 2 will send a flow control signal to switch 1, disrupting the flow on the uncongested path from A to X. Virtual LANs provide a more flexible way to manage switched LANs.



3.18

FIGURE

Congestion at Z causes flow control on link L, which interferes with un congested flow from A to X.

### Virtual LAN

When a bridge connects two LANs, traffic local to each LAN segment can be transmitted simultaneously. However, only computers that are close together can be attached to the same physical LAN segment. Ethernet switches permit a logical grouping of computers and switch ports into different *virtual LANs* or VLANs. A VLAN does not require physical proximity and can span several interconnected buildings. VLANs are used to group computers sharing common servers or a common organizational function, or to provide security.

A VLAN is a subset of computers and ports within a single switched LAN domain. The subset is defined according to administrative rules. In the simplest case, the LAN administrator statically assigns ports to a VLAN. In more elaborate arrangements a VLAN is defined using packet filtering. Attributes of a packet such as MAC addresses are examined to determine VLAN membership, and a filtering table is developed for each switch. Table entries are compared with the packets filtered by the switch, and the switch takes the appropriate forwarding action. Packet filtering, however, requires an additional layer of processing as compared with static port assignment. This processing may involve layer 3 (network layer) information.

A packet broadcast on a bridged network of LANs is transmitted to all the attached computers. In this sense, all these computers form a single *broadcast domain*. VLANs can be used to segment the broadcast domain.

ATM switches can also be configured to create VLANs, as explained in Chapter 6.

## 3.3

### TOKEN RING (IEEE 802.5)

The IEEE 802.5 standards specify layers 1 and 2 of a family of *token ring* networks. These networks transmit at 4 Mbps or 16 Mbps. (Standards are

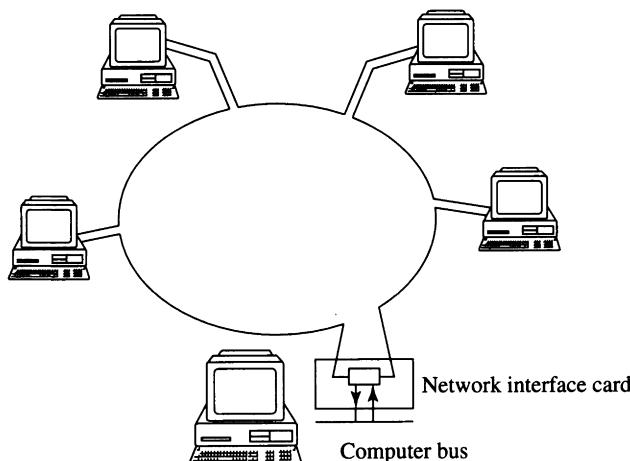
being developed for 100-Mbps transmission speed.) These networks have the advantage that, unlike in Ethernet networks, each node is guaranteed to be allowed to transmit before a specific time. Also, the token ring networks are more efficient than Ethernet networks under high load.

### 3.3.1 Physical Layer

In a token ring network, the nodes are connected into a ring by point-to-point links. (See Figure 3.19). A network interface has two possible configurations: repeater and open. In the repeater configuration, the interface repeats the incoming signal on the outgoing link with a delay of a few bit transmission times. At the same time, the interface copies the signal for the computer. In the open configuration, the interface transmits on the outgoing link and listens on the incoming link. The transmission rate is 4 Mbps or 16 Mbps, as already mentioned. As with Ethernet, signals may be transmitted over a variety of cabling arrangements, including UTPs.

### 3.3.2 MAC

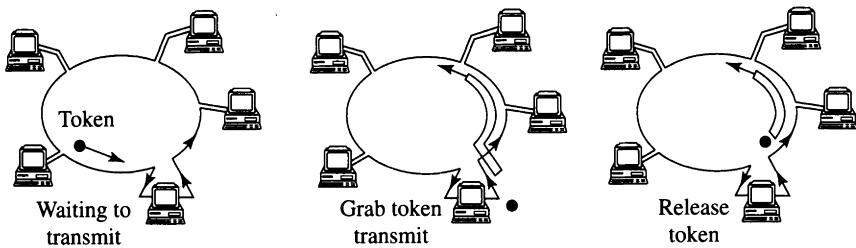
The frame format is similar to that of Ethernet packets (Figure 3.11), except that it uses an ending delimiter instead of a length indication. The token is a



3.19

FIGURE

Layout of a token ring network. The computers are attached by unidirectional point-to-point links around a ring.



3.20

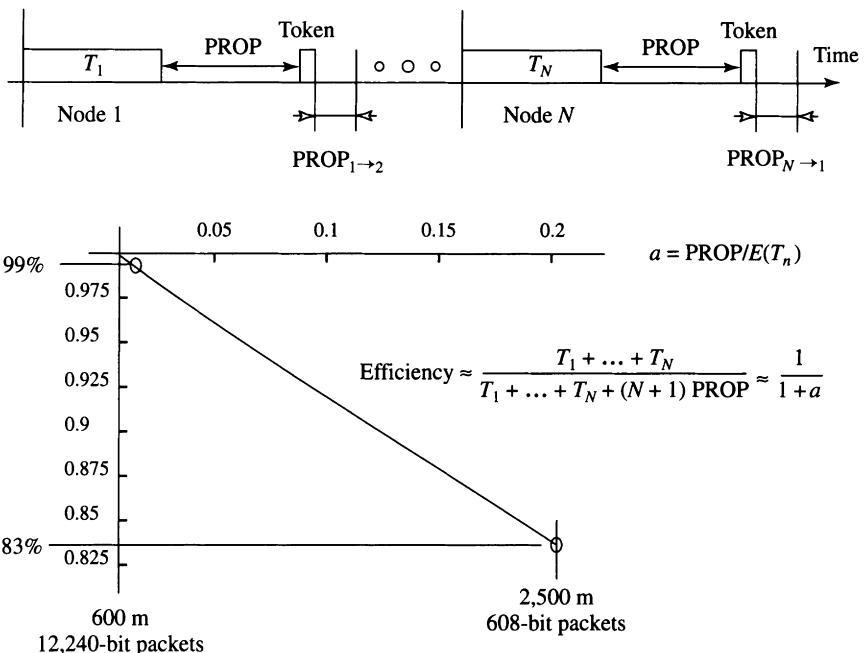
FIGURE

Steps in the transmission of a packet when the computers use the release after transmission token-passing protocol.

3-byte frame that consists of a start of frame, an access control, and an ending delimiter, each 1 byte long. The access control field indicates that the 3-byte frame is a token that any station may grab and not a packet.

The transmissions proceed in one direction along the ring. Figure 3.20 shows the sequence of events when a node wants to transmit a packet: the node waits for the token, then transmits some of its packets and releases the token. We call this version of the MAC protocol, where a node releases the token right after it finishes transmitting its packets, *release after transmission*. The 16-Mbps token ring networks use this protocol. In a 4-Mbps token ring network, a node that transmits waits until it has completely received its last packet before releasing the token. We call this version *release after reception*. The standard specifies that a node can hold onto the token and transmit for up to some time, called the *token holding time* (THT), before releasing the token. A typical value of THT is 10 ms.

We use Figure 3.21 to analyze the efficiency of the release after reception protocol. Assume that there are  $N$  nodes on a token-passing ring. We define  $T_n$  to be the time during which node  $n$  transmits when it gets the token, before it releases the token. Thus,  $T_n$  can range from 0 to THT. We assume that all the nodes want to transmit, so that  $T_n > 0$  for  $n = 1, \dots, N$ . At time 0, the first node starts transmitting a packet. At time  $T_1$  the first node has transmitted its packets. The last packet has completely returned to the first node PROP seconds later, where PROP is the propagation time of a signal around the ring. Therefore, at time  $T_1 + \text{PROP}$  the first node starts transmitting the token. A short time later, the first node finishes transmitting the token, which reaches the second node after a propagation time designated by  $\text{PROP}_{1 \rightarrow 2}$ . Node 2 then goes through the same sequence of steps node 1 did, and so do the other nodes, one after another. Eventually, the token comes back to node 1. The efficiency



3.21

FIGURE

Timing diagram for the release after reception token-passing protocol when all the computers have packets to transmit.

of the token ring is the fraction of time that the nodes transmit packets. The figure shows that the efficiency is approximately equal to  $1/(1 + a)$  where  $a = \text{PROP}/E(T_n)$ . In this expression,  $E(T_n)$  is the average duration of a node transmission. The figure also shows representative values of the efficiency, assuming that the nodes transmit a single packet of a fixed size. As we can see, the efficiency of a typical token ring network is more than 90%.

One can modify Figure 3.21 to study the release after transmission protocol used by the 16-Mbps token ring network and, with a similar analysis, show that its efficiency is approximately  $(1 + a/N)^{-1}$ , which approaches 100%.

### 3.3.3

### LLC

The logical link control sublayer of IEEE 802.5 networks is the same as in IEEE 802.3 networks.

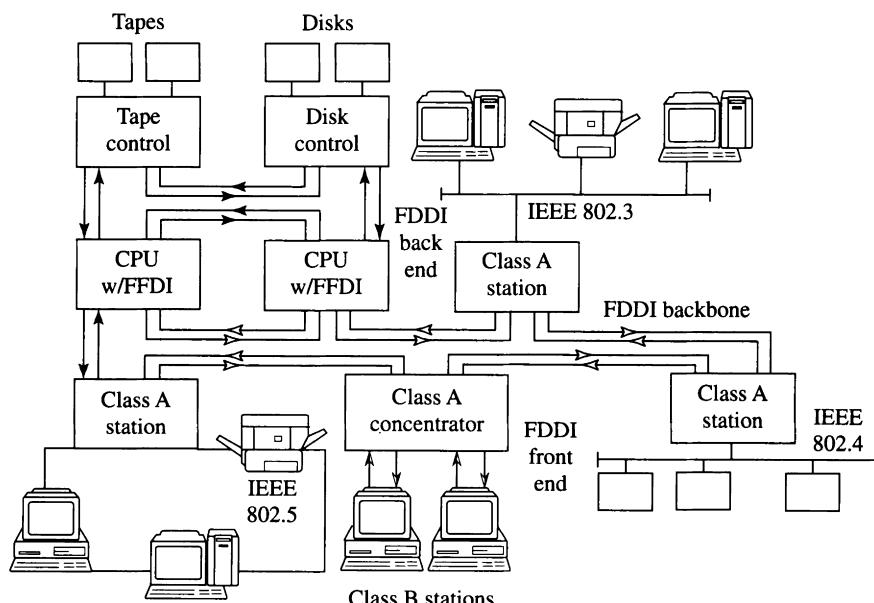
### 3.4

### FDDI

The Fiber Distributed Data Interface (FDDI) (see Figure 3.22) is an ANSI (American National Standards Institute) standard for a 100-Mbps network, published in 1987. Until recently, FDDI was the preferred technology for connecting LANs. Gigabit Ethernet and ATM switches, which operate at higher speeds, are expected to displace FDDI.

FDDI connects up to 500 nodes with optical fibers, in a dual ring topology. The distance between adjacent nodes cannot exceed 2 km when multimode fibers and LEDs are used. Longer separation is possible with single-mode fibers and laser diodes. The maximum length of the fibers is 200 km. Because of this length, FDDI networks are used to interconnect computers within a campus. Many vendors supply FDDI hardware and software for workstations.

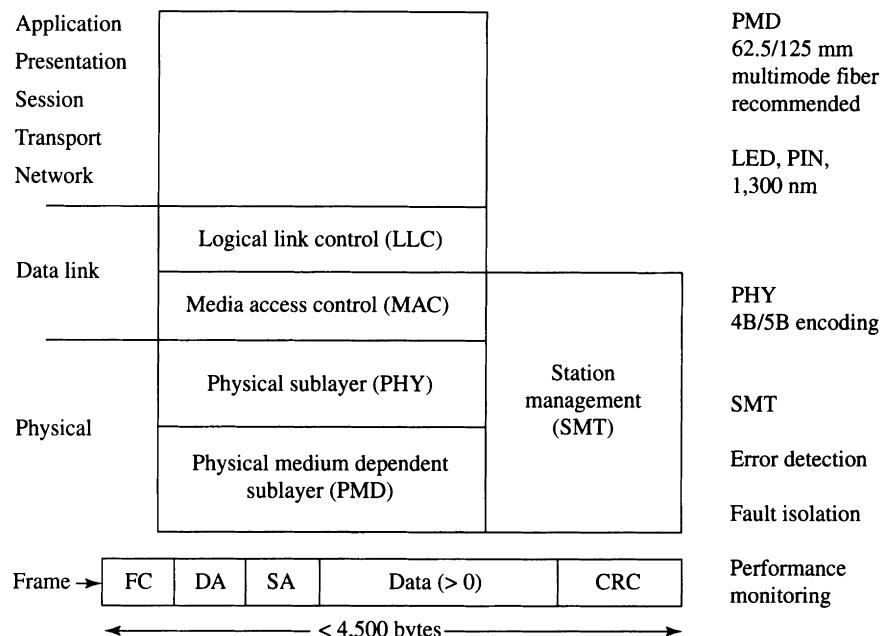
The figure shows FDDI networks that connect workstations to file servers and printers, or workstations together, or terminals and terminal emulators to workstations.



3.22

FIGURE

An FDDI dual ring network can support 500 stations with a total distance of 200 km and up to 2 km between adjacent stations. Stations are connected by 100-Mbps optical fiber, and a timed-token MAC protocol is used.



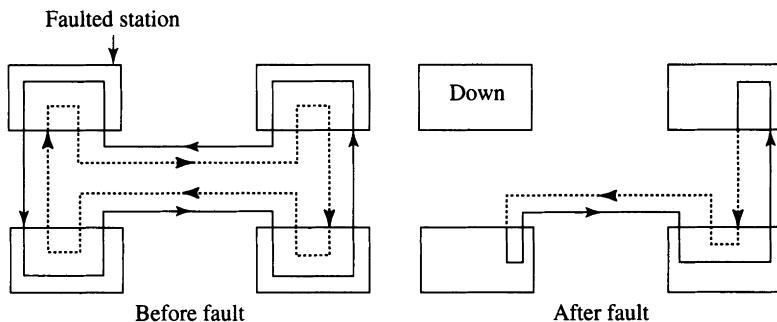
3.23

**FIGURE**

The FDDI standards specify the MAC sublayer and the physical layer of the protocol stack.

As indicated in Figure 3.23, the FDDI standards specify the MAC sublayer and the physical layer. The physical layer itself is divided into two sublayers. The standards also specify the station management (SMT) protocols. The PMD (physical medium dependent) sublayer specifies the fiber to be used as well as the optical sources and detectors. The specifications of PMD are summarized in Figure 3.23. It should be noted that vendors make alternative PMD products available. For instance, twisted pairs can be used to connect stations separated by less than 100 m.

The PHY (physical) sublayer specifies that the stations must use the 4B/5B encoding. With this encoding, the transmitter groups the bits by 4 and converts each 4-bit word into a 5-bit word specified by the encoding table. The 16 words of 5 bits that the encoding uses were chosen so that the resulting optical signal contains enough transitions to keep the receiver synchronized. Note that with this encoding, the 100-Mbps data rates result in a raw bit stream of 125 Mbps on the fibers. If the transmitters had used Manchester encoding, the optical signal would have transitions at 200 MHz, necessitating more expensive electronics.



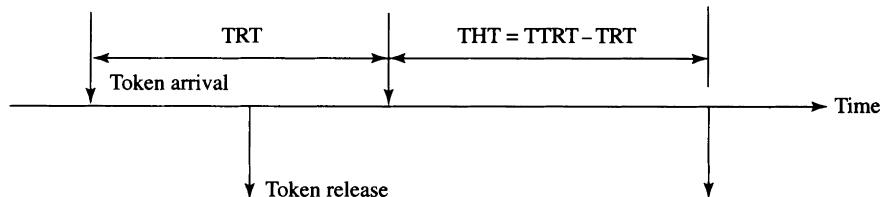
3.24

FIGURE

When a fault is detected, the rings are reconfigured to isolate the faulted station.

The SMT must detect errors and isolate a fault on the ring, such as a failure of a station or link on the ring. Figure 3.24 illustrates how the dual ring is reconfigured as a single ring after the fault has been isolated. In addition, the SMT monitors the performance of the network. The MAC of FDDI specifies that the frames have a maximum length of 4,500 bytes. (The frame structure is illustrated in Figure 3.23.) The MAC uses a timed-token protocol. This protocol is similar to the token-passing mechanism of IEEE 802.5, except for the timing feature, as we explain next.

Figure 3.25 helps to explain the MAC protocol when the stations transmit only asynchronous traffic. Assume that the stations are initially idle, that is, they have no packet to transmit. A token, which is a packet with a specific bit pattern, travels around the ring. Each station has two timers: TRT or token rotation time timer, which counts up, and THT or token holding time timer,



3.25

FIGURE

The timed-token protocol guarantees that each station will get a chance to transmit in less than TTTRT. TTTRT is the target token rotation time.

which counts down. When a station has a packet to transmit, it waits until it gets the token. When the station gets the token, it does the following:

1. Grabs the token.
2. Sets  $THT = TTRT - TRT$ . ( $TTRT$  or target token rotation time is set by the network manager.)
3. Resets  $TRT = 0$ .
4. Transmits packets until  $THT = 0$  or there is no packet left.
5. Releases the token.

Figure 3.25 shows for a particular station two successive token arrival and release times. Suppose that the time,  $TRT$ , between successive arrivals is less than  $TTRT$ . The figure shows that in that case the time between successive token releases is also less than  $TTRT$ . But this is the time between successive arrivals for the next station. By repeating the argument, we conclude that every station will wait for time at most  $TTRT$  for a token arrival. If we assume that a station may complete transmitting its current packet even when  $THT = 0$ , then this argument must be slightly modified to conclude that each station waits at most  $TTRT + TRANS$ , where  $TRANS$  is the longest packet transmission time.

The transmitting station must remove its own packet from the ring: the station waits until it receives the packet that it transmitted, that is, until it reads its own physical address as the source address of the packet, and it then removes the packet by transmitting “idle” symbols instead of repeating the packet.

Actually, the MAC protocol provides for two types of traffic: asynchronous and synchronous. As will be explained, the stations get to transmit their synchronous traffic at least every 2  $TTRT$  seconds. For instance, if the stations agree on a value  $TTRT = 20$  ms, then the stations that transmit synchronous traffic, say voice, get to transmit at least every 40 ms. If the voice is encoded into a 64-Kbps stream, then the stations need only be able to buffer  $40 \times 10^{-3} \times 64 \times 10^3 = 2,560$  bits of voice.

We now explain how the protocol accommodates synchronous traffic. The stations first request permission to transmit synchronous traffic. The network eventually decides which stations can transmit synchronous traffic, and it allocates a fraction of  $TTRT$  to each of those stations. The fractions add up to one. The protocol works as follows. When a station that can transmit synchronous traffic gets the token, it does so for up to the fraction of  $TTRT$  that it was allocated. It transmits asynchronous traffic as before, using the previously de-

scribed protocol. When the stations use this protocol, they get the token at least once every 2 TTRT seconds.

As was explained in Figure 3.25, the MAC results in a bounded medium access time that is suitable for synchronous traffic. Note, however, that a station does not access the medium exactly at periodic times. Thus, the FDDI MAC does not implement an isochronous transmission facility. It can also be shown that the FDDI MAC provides a fair allocation of the bandwidth to the different stations for asynchronous traffic. (A fair allocation is one in which every station has the same probability of transmission access. The fairness of the FDDI MAC is not quite obvious from our description of the protocol.) Moreover, the FDDI MAC protocol is very efficient because the overhead that it imposes does not increase when the stations have many packets to send. FDDI was designed to support multimedia connections where the stations exchange video, audio, text, and data. FDDI is being used to interconnect LANs.

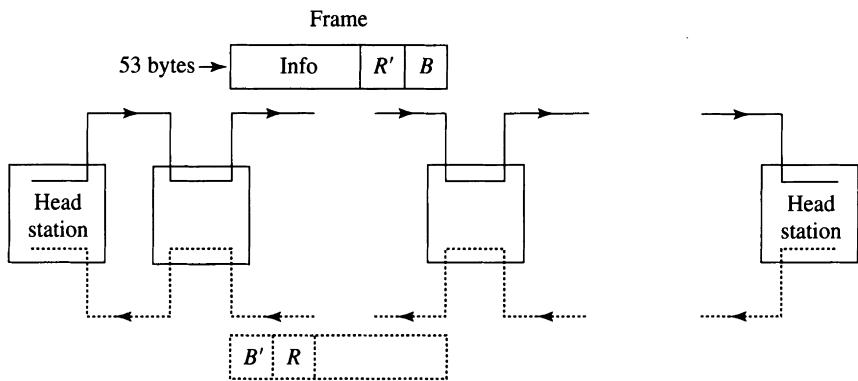
In summary, by using a timed-token mechanism instead of an untimed token-passing protocol or CSMA/CD, FDDI guarantees a bounded medium access time and is therefore suitable for synchronous transmission services in addition to asynchronous transmissions. Thus, the faster physical layer of FDDI increases the throughput, and the timed-token mechanism of its MAC enables the transport of constant bit rate traffic. Moreover, FDDI is designed to be reliable against link or node failures. In 1999, it appears that FDDI is on the way out and is being replaced by switched Ethernet and, in some cases, by ATM as an interconnection technology for Ethernets.

### 3.5

### DQDB

The Distributed Queue Dual Bus (DQDB), illustrated in Figure 3.26, is the IEEE 802.6 standard for a MAN (metropolitan area network). The figure shows the topology of DQDB. Each station is attached to two unidirectional buses. The word *bus* is a misnomer, because the connections in each direction are implemented by a sequence of point-to-point links instead of a bus as in Ethernet or a token bus. (There are few DQDB vendors; given the popularity of FDDI and ATM, it is unlikely that DQDB will be widely deployed. Its main interest is that the DQDB protocol is used in the subscriber-network interface for SMDS.)

The DQDB MAC protocol is a clever way to regulate access to the medium as if all stations placed their packets in a single queue that is served on a first-come, first-served (FCFS) basis. Such a first-come, first-served protocol would



3.26

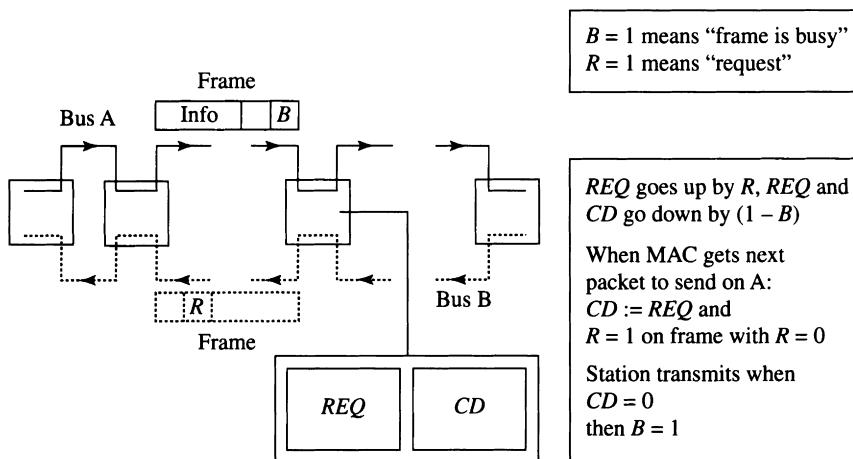
FIGURE

In the DQDB network, frames are generated back to back by the left head station with  $R' = B = 0$ ; a station can use a frame if there is no pending request to its right. The operation of the lower bus is similar.

be the fairest possible. However, it cannot be achieved perfectly because the queues are distributed in the different stations and no station knows exactly when the other stations got packets to transmit. We will see how the DQDB protocol approximates that FCFS behavior. A station wanting to transmit to another station situated to its right must use the upper bus, and it must use the lower bus to transmit to stations on its left. The operations of the two buses are identical; how the nodes transmit on the upper bus is shown in Figure 3.27.

Fifty-three byte frames are generated back to back by the head stations. Each frame has two special control bits: the busy bit  $B$  and the request bit  $R$ . The head stations generate idle frames: the frames from the left head station have  $R' = B = 0$ , and frames from the right head station have  $R = B' = 0$ . When a station wants to transmit and when the protocol described below allows it to transmit, the station uses the next idle frame to transmit its own frame, setting the busy bit to 1. Each station copies each frame and retains copies addressed to itself. The final head station removes the frame from the bus.

We now explain the MAC protocol. When a station  $S$  wants to transmit on the upper bus, it must first reserve a frame. To reserve a frame on the upper bus,  $S$  waits until it sees a frame on the lower bus with its request bit  $R = 0$ .  $S$  then sets bit  $R = 1$ . When that frame propagates, the stations to the *left* of  $S$  learn that one station to their *right* has a packet to transmit on the upper bus. By counting these requests, every station can keep track of the total number of packets that stations to its right want to transmit. More precisely, when station  $S$  gets a packet to transmit, it knows how many frames have been reserved by



3.27

The figure explains the operation of the DQDB protocol.

FIGURE

stations to its right.  $S$  stores that number in two counters:  $CD$  (count-down) and  $REQ$  (request). The DQDB protocol specifies that the stations must defer to their right. That is, station  $S$  cannot use an idle frame that comes by on its upper bus until all the  $CD$  reservations made by stations to its right have been serviced.

In order to implement this protocol, every time  $S$  sees an idle frame (indicated by  $B = 0$ ) go by on its upper bus, it decrements  $REQ$  by one; and every time  $S$  sees a reservation (indicated by  $R = 1$ ) on the lower bus, it increments  $REQ$  by one. So at each time,  $REQ$  is the number of outstanding requests from stations to the right of  $S$ . When  $S$  itself gets a packet to transmit, it loads the  $CD$  counter by the current value of  $REQ$ ,  $CD := REQ$ . (This is the number of outstanding requests from stations to the right of  $S$  at the time it received a packet.) It then decrements  $CD$  by one each time an idle frame goes by on the upper bus. As soon as  $CD$  reaches zero,  $S$  knows that all the reservations to its right that were placed before it got its packet to transmit have been serviced. Station  $S$  is then allowed to use the next idle frame to transmit its own packet.

The DQDB MAC protocol is very efficient. Unlike Ethernet, there is no loss of capacity due to collision. Unlike token ring, an idle frame is continuously generated by the head station. If there always are stations with packets to transmit in both directions, utilization will be 100%. However, the protocol is not perfectly fair because its topology is not symmetric. For instance, the

leftmost station must transmit all its packets on the upper bus, and it must defer to all the other stations to its right. By contrast, a station in the middle transmits half its packets on each bus and defers to only half the stations on each bus. To correct this unfairness, the standard specifies that each station be allocated an individual parameter  $F$ .  $F$  specifies the number of successive frames that the station can use to transmit. The network manager can select these parameters so that the resulting utilizations of the buses by the different stations are comparable.  $F$  is called the *bandwidth balancing* parameter. The IEEE 802.6 standard specifies only the MAC protocol of DQDB. The standard also provides for different traffic priorities. Priorities are implemented by having distinct  $B$  and  $R$  bits and distinct counters for different priorities.

The networks considered above are used as local area networks that connect nearby computers or as campus networks that connect computers or LANs in nearby buildings. We now describe two wide area packet-switched networks, Frame Relay and SMDS. These networks are used to connect computers or LANs across a public switched network.

### 3.6

### FRAME RELAY

Frame Relay is a connection-oriented data transport service for public switched networks. The Frame Relay protocols are a modification of the X.25 standards. Both X.25 and Frame Relay specify the lowest three OSI layers for virtual circuit networks. Frame Relay standards are specified by the International Telecommunications Union (ITU) and ANSI, beginning in 1990.

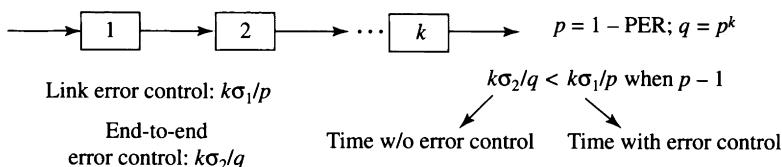
X.25, introduced in 1974, was designed to operate with noisy transmission lines. Accordingly, the link level protocol of X.25 (called LAPB for Link Access Procedure B) performs error detection and recovery using the Go Back N protocol with a window size of 8 to 128. LAPB also provides for some link level flow control by enabling a receiver to stop the sender temporarily by sending it a control frame. The network layer of X.25 specifies that up to 4,096 virtual circuits can be set up on any given physical link. An end-to-end window flow control can be implemented along each virtual circuit independently of the link level flow control.

Frame Relay is simpler than X.25. It is designed to take advantage of links with a higher transmission rate and small bit error rate. (X.25 is intended to work with 64-Kbps links; Frame Relay works with 56-Kbps, 1.5-Mbps, and higher-speed links.) The main difference with X.25 is that Frame Relay does not control errors at the link level. Instead, error control and recovery are done

by higher layers. Consequently, the packet or frame processing time at each node is smaller than for X.25. Moreover, the transmissions on a link are not slowed down as they would be by the Go Back N protocol of X.25 when its window has been transmitted and the sender waits for the acknowledgments to come back before resuming the transmissions. Thus, Frame Relay is a virtual circuit service which does not provide reliability.

We will explain why it is advantageous to replace link level error control by end-to-end control when the bit error rate is small. We will then show why Go Back N slows down transmissions when the bandwidth-delay product of the link exceeds the window size. These two observations justify the superiority of Frame Relay over X.25 for higher-speed, low-error links. Since Frame Relay is simpler than X.25, most vendors of X.25 equipment provide software to run Frame Relay on their switches. Frame Relay is a popular means to interconnect networks.

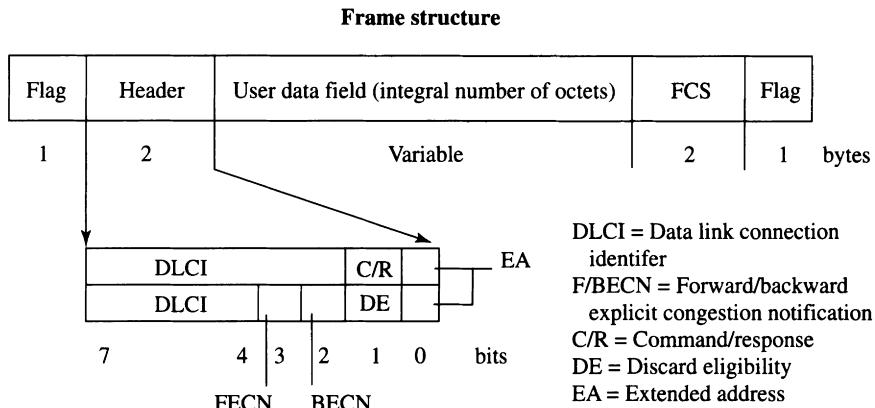
For the first observation, consider the virtual circuit connection model of Figure 3.28. The connection goes through  $k$  nodes. Each node takes a fixed time  $\sigma_1$  to process and transmit each frame. Assume that each transmission from one node to the next corrupts the frame with probability  $1 - p$ . If errors are controlled by the link and each frame is retransmitted until it is successfully received, then each frame must be transmitted  $1/p$  times on average before being successfully received. Consequently, the average transmission time of the frame by the  $k$  nodes is equal to  $k\sigma_1/p$ . If no link level error control is performed, then the node processing and transmission time is assumed to be  $\sigma_2 < \sigma_1$ . (The difference in practice is large:  $\sigma_1 \sim 50$  ms compared with  $\sigma_2 \sim 3$  ms.) The total transmission time by the  $k$  nodes is now  $k\sigma_2$ . However, this end-to-end transmission time must take place  $1/q$  times on average, where  $q$  is the probability that no transmission corrupts the packet, that is,  $q = p^k$ . Since  $k\sigma_1/p > k\sigma_2/q$  whenever  $p$  is sufficiently close to 1 (i.e., the bit error rate is sufficiently small), the end-to-end error control becomes faster than the link level error control.



3.28

FIGURE

Frame Relay provides faster processing of packets because it does no link error control.



3.29 Frame format of Frame Relay.

**FIGURE**

In section 2.6.3 we defined the efficiency of a transmission protocol as the fraction of time the transmitter is sending new packets. We showed that the efficiency of the Go Back N protocol is

$$\text{Efficiency} = \min \left\{ \frac{N \text{ TRANS}}{\text{TRANS} + \text{ACK} + 2\text{PROP}}, 1 \right\},$$

where TRANS is the packet transmission time, ACK is the time to transmit the acknowledgment, and PROP is the propagation time between sender and receiver. Thus to achieve Efficiency = 1 we must have  $N \text{ TRANS} \geq \text{TRANS} + \text{ACK} + 2\text{PROP}$ . Neglecting ACK, this implies  $N$  should be at least  $2\text{PROP}/\text{TRANS}$ , that is, the window should be large enough to “fill up the pipe.” For example, suppose the end-to-end distance is 5,000 km, so PROP equals  $5 \times 5,000 = 25 \mu\text{s}$ . For a 1,000-byte packet and a transmission speed of 50 Mbps, TRANS =  $160 \mu\text{s}$ . This gives a window size of about 300 packets.

The frame format is shown in Figure 3.29. The 2-byte header contains address information (DLCI and EA, explained below) for routing, congestion-control information (F/BECN and DE, also explained below) for notification and enforcement, and the C/R bit whose usage is application-specific. The frame check sequence FCS is a 16-bit CRC for error detection: erroneous frames are discarded and are not retransmitted by the network. The standard specifies the use of Permanent Virtual Circuits (PVCs) for connections.

A PVC is a fixed route assigned between two users when they subscribe to a Frame Relay service. A PVC is identified at the network interface by a 12-bit *data link connection identifier* (DLCI). DLCIs specify and distinguish separate

connections across an access link and therefore can be used to multiplex several connections. The DLCI field allows for 1,024 PVCs per access link. Of these, about 1,000 can be assigned to users, and the rest are reserved for control purposes. The header may be extended to 4 bytes to accommodate more DLCIs. The EA (extended address) bit is used for that purpose: EA = 0 indicates that the next byte is also an address byte; EA = 1 indicates the last address byte.

Frames are discarded at a node or switch when erroneous or when buffers overflow. To reduce buffer overflow, the switch can exercise flow control as follows. When a switch experiences some congestion, it notifies the sources and destinations of all the active PVCs passing through the node. This is done by setting the FECN (forward explicit congestion notification) bit in user frames going in the forward direction to inform the destination or the BECN (backward explicit congestion notification) bit in user frames going in the reverse direction to inform the source. FECN may be used by destination-controlled flow-control protocols, whereas BECN may be used by source-controlled flow-control protocols. The Frame Relay standard, however, does not define congestion, nor does it specify how users should respond to it.

The DE (discard eligibility) bit may be set by users to indicate low-priority frames such as some audio or imaging frames with less significant information. Note, however, that compressed video is more sensitive to losses so that such packets might not have the DE bit set. The DE bit may also be set by a network node. The network would preferentially discard frames with DE = 1 when necessary to alleviate congestion. (ATM packets also incorporate a 1-bit priority; see Chapter 6.) DLCI = 1,023 is a PVC reserved for communication between the user and the network. The user and the network periodically exchange “keep alive” messages on that PVC. A user could also poll the network on that PVC, at which point the network would report all active DLCIs on that access link and their traffic parameters such as CIR (explained below). That PVC can also be used for flow control, especially when there is too little traffic through a congested node in the reverse direction for a timely notification of congestion by the BECN.

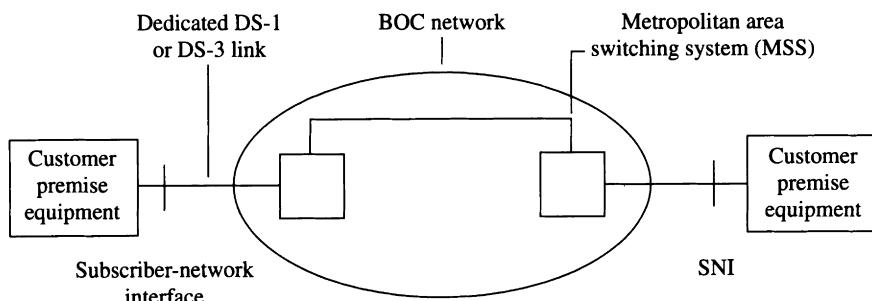
At subscription time, each DLCI is assigned three parameters ( $T_c$ ,  $B_c$ ,  $B_e$ ) for traffic shaping. These parameters are used as follows. Time is slotted into intervals of duration  $T_c$ . The network guarantees transport of  $B_c$  bytes of data in each interval. This guarantees a “committed information rate”  $CIR = B_c/T_c$ . If the user injects more than  $B_c$  bytes across the user-network interface in an interval, the network may admit the first  $B_e$  bytes of excess data with their DE bits set. Further frames in that interval may be discarded. The DLCI is guaranteed a long-term bandwidth of  $CIR$  and a maximum burst size of  $B_e$ . This traffic-shaping scheme regulates the input load to the Frame Relay

network, thus reducing the likelihood of congestion. The scheme may be implemented by using a leaky bucket for each PVC at the network entrance. (The leaky-bucket scheme is described in section 3.7.) Traffic-shaping schemes are discussed in Chapters 8 and 9.

In summary, Frame Relay is an improvement over X.25 networks, taking advantage of better transmission links by streamlining the X.25 protocol. However, its switches lack the capability to reserve resources for individual connections, and so Frame Relay is unsuitable for applications that require guaranteed delay. It is interesting to note, nevertheless, that many of the developments used to differentiate service quality occur almost simultaneously in Frame Relay, SMDS, and ATM. Of these three designs, ATM will be the most successful in offering differentiated services. The Frame Relay and ATM Forums are developing interworking standards and specifications.

### 3.7 SMDS

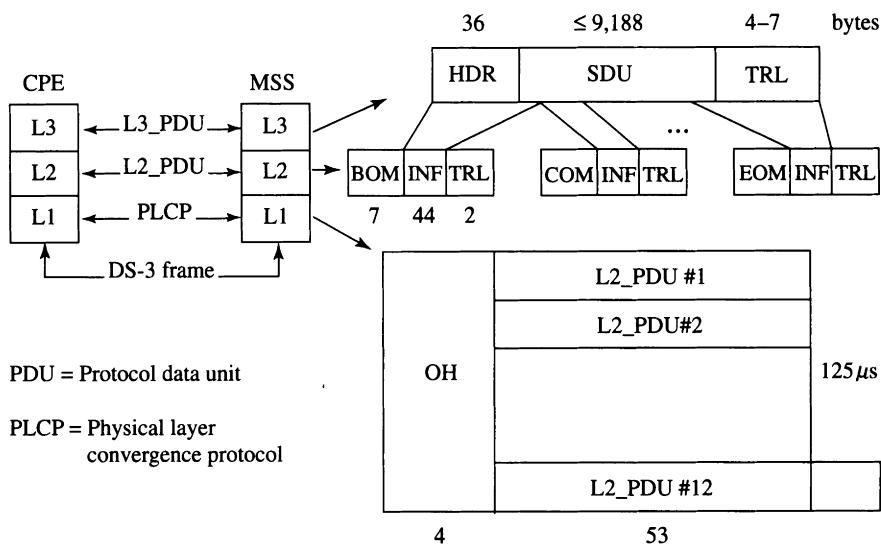
Switched Multimegabit Data Service (SMDS) is a public switched connectionless data transport service, defined by Bellcore. Beginning in 1992, SMDS has been offered by regional Bell operating companies (BOCs) at DS-1 (T-1) access speed (1.54-Mbps line rate corresponding to 1.17-Mbps data rate). DS-3 (T-3) access speed (45-Mbps line rate or 34-Mbps data rate) is also available. SMDS will later be offered at much higher rates over a SONET network. In some regions, SMDS is growing rapidly. Subscriber equipment is connected through the BOC network. Access to that network is via dedicated access lines. (See Figure 3.30). The standard specifies the protocol at the subscriber-network interface.



3.30

The SMDS network and interfaces.

FIGURE



3.31

The SMDS protocol stack and frame structure.

FIGURE

The SMDS protocol roughly corresponds to the first three OSI layers. It is divided into three levels. Level 1 provides the physical interface to the digital network. Level 2 defines a cell structure similar to ATM cells and performs error detection. Level 3 handles addressing and routing. (The 53-byte cell structure will permit a migration path for SMDS to ATM, as we will see in Chapter 6.)

Figure 3.31 shows the formats of the protocol data units (PDUs) at levels 2 and 3. User data, up to 9,188 bytes, is encapsulated in an L3\_PDU. The L3\_PDU overhead contains the full source and destination addresses, the L3\_PDU length and, optionally, a CRC for detecting L3\_PDU errors. Each address is specified by 15 BCD (binary coded decimal) digits. (The addressing scheme is identical to the North American telephone numbering system.) The total L3\_PDU overhead may vary from 40 to 43 bytes. An L3\_PDU is fragmented into a sequence of L2\_PDUs. Lastly, the function of the Physical Layer Convergence Protocol or PLCP is to place one or more L2\_PDUs into a frame of the physical link. As an example, the figure shows how 12 L2\_PDUs are assembled into one 125- $\mu$ s DS-3 frame.

Each L2\_PDU is 53 bytes long and contains 44 bytes of L3\_PDU payload. If the payload is less than 44 bytes, it is padded to make a 53-byte L2\_PDU. The 2-byte trailer contains a payload length to indicate the size of the padding

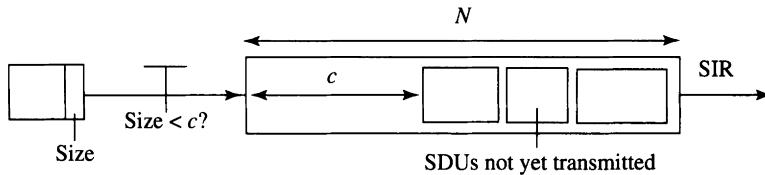
and a payload CRC to detect L2\_PDU errors. The 7-byte header contains a 10-bit MID (message identifier), a 2-bit segment type indicating beginning (BOM), continuing (CM), or end (EOM) of message, and a 4-bit sequence number for reassembly of L3\_PDUs at the destination. Before any L3\_PDU is transmitted, it is assigned a MID, which is borne by each of its L2\_PDUs. The MIDs should be unique among all L3\_PDUs being simultaneously transmitted. The sequence of L2\_PDUs belonging to the same L3\_PDU is ordered using the sequence number. The unique MID allows the destination to collect all L2\_PDUs belonging to the same L3\_PDU. The sequence number makes correct reassembly possible even when the L2\_PDUs arrive out of order. (The standard specifies that the L2\_PDUs must be delivered in order.)

Since an L2\_PDU does not contain the full destination address, it cannot be routed individually. A simple connection-oriented implementation of the SMDS service is as follows. A virtual circuit is set up to transfer each L3\_PDU, identified by its MID. All L2\_PDUs then follow the same path using the MID as the virtual circuit identifier. They can be reassembled at the destination using the L3\_PDU length field. This method ensures delivery of the packets in the correct order and hence does not require the sequence number. With this implementation, SMDS provides a datagram service for L3\_PDUs using a connection-oriented L2\_PDU delivery service.

Unlike previous data networks, SMDS offers several service-quality levels defined by service parameters, chosen at time of subscription. We discuss three parameters: address screening, limit on number of simultaneous packets, and information rate. Address screening means that packets may be received from or delivered to only a specified list of destination and source addresses. (Address screening is a security device.)

To understand the second parameter, observe that the network may interleave L2\_PDUs belonging to different L3\_PDUs and intended for the same destination. Before forwarding these L2\_PDUs, the switch must deinterleave them and forward the L2\_PDUs of one L3\_PDU together. The switch must buffer these L2\_PDUs to do this deinterleaving. A limit on the number of simultaneous packets will place a limit on the needed buffer size. The maximum information rate is specified by two parameters, the sustained information rate, SIR, and the maximum burst size,  $N$ .

The restriction on the information rate is implemented by the leaky-bucket scheme of Figure 3.32. A buffer or “bucket” of size  $N$  bits is served (read out) at a constant rate of SIR bps. A packet is accepted into the buffer only if its size is smaller than the size,  $c$ , of the free buffers; otherwise the packet is blocked. It can be seen that if the subscriber submits packets of size  $n_1$  at time  $t_1$ , size  $n_2$



3.32

FIGURE

The leaky-bucket scheme guarantees that a user's traffic will not have a sustained information rate higher than SIR or a burst of size larger than  $N$ .

at time  $t_2$ , and so on, then no packets will be blocked provided that for every  $i < j$ ,

$$\sum_i^j n_k \leq (t_j - t_i) \times \text{SIR} + N.$$

This formula makes precise the restriction that the subscriber's traffic on average cannot exceed SIR bps, and it cannot have a burst of more than  $N$  bits.

### 3.7.1 Internetworking with SMDS

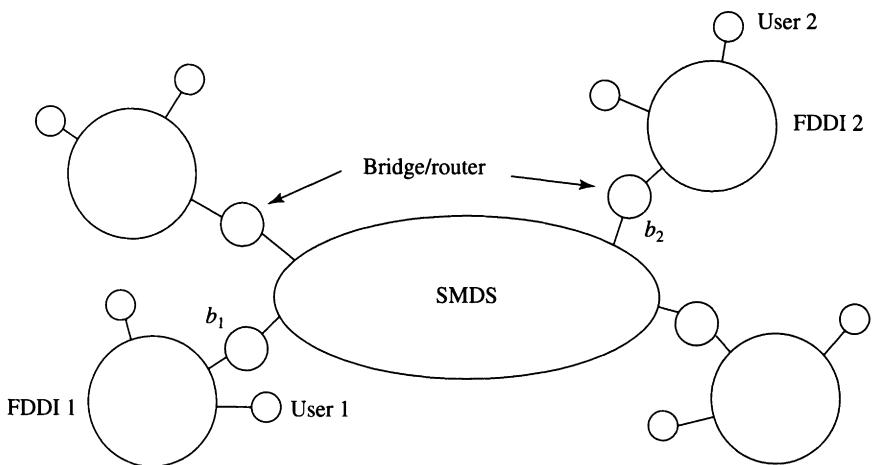
SMDS and Frame Relay are used to interconnect local area networks. We conclude by describing the functions that must be carried out in order that users connected to different FDDI LANs can transparently interconnect over an SMDS network. (The functions are similar if Frame Relay is used.) Figure 3.33 shows user 1 on one FDDI ring who wishes to send a packet to user 2 on another ring.

Assume first that user 1 knows the MAC address of user 2. In that case user 1 places a frame on FDDI 1 with destination address of user 2. Bridge  $b_1$  on that ring must

1. copy that frame;
2. "convert" it into an L3\_PDU, address it to station  $b_2$  (using its SMDS address), and submit it to the SMDS network.

Note that bridge  $b_1$  is a station on FDDI 1 and on the SMDS network. In order to "submit" the L3\_PDU, it must go through the SMDS protocol stack, fragmenting the L3\_PDU into L2\_PDUs, and then organizing them for transmission into L1\_PDUs as in Figure 3.31.

When bridge  $b_2$  receives these L1\_PDUs, it assembles them into the L3\_PDU. Subsequently,



## 3.33

FDDI networks can be interconnected using SMDS.

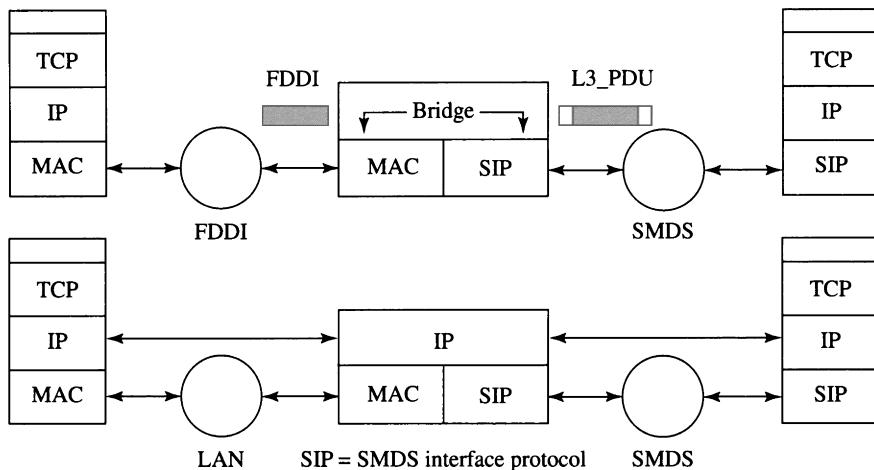
**FIGURE**

1. bridge  $b_2$  must “convert” the L3\_PDU back into the original FDDI frame, place it on FDDI 2, and then remove it from that ring when it returns (note that the source address on the frame is that of user 1 and not  $b_2$ );
2. user 2 copies this FDDI frame.

Bridge  $b_2$  is on FDDI 2 and on the SMDS network.

Neither user 1 nor user 2 knows that they are not on the same ring—the FDDI frame is forwarded transparently. Bridge  $b_1$  must carry out two functions: address conversion and frame conversion. When  $b_1$  reads the FDDI frame addressed to user 2, it must recognize that user 2 is on a “remote” ring that can be reached via bridge  $b_2$ . This address conversion is carried out with the help of two address tables. The first has entries of the form: (remote user MAC address, remote bridge SMDS address). The second is a list of all MAC addresses on its own ring. The bridge must then convert the FDDI frame into an L3\_PDU. The most common way to do this is to encapsulate the entire FDDI frame as a payload in an L3\_PDU. This is shown in the top panel of Figure 3.34. We will shortly see how the address table can be constructed.

After bridge  $b_2$  assembles the L3\_PDU, it must decapsulate it and recover the FDDI frame. From its address table,  $b_2$  recognizes that the destination address, user 2, is on its “own” ring. It then places the frame on the ring, and removes it when it returns.



3.34

Internetworking with SMDS.

FIGURE

The two address tables, one with remote MAC and SMDS bridge addresses, the other with MAC addresses on its own ring, can be built up as follows. Each time  $b_1$  receives an L3\_PDU, it notes the SMDS address of the sending bridge and, from the encapsulated FDDI frame, it obtains the MAC address of the source. In this way it builds the first table. The second table is built simply from the source address on each FDDI frame on its own ring.

In case user 1 uses the IP address of user 2, the stations  $b_1$  and  $b_2$  will have to be routers. This requires additional functions suggested in the lower panel of Figure 3.34.

### 3.8

### SUMMARY

The packet-switched networks studied above show a steady advance in speed, connectivity, delay, and flexibility or ability to accommodate more kinds of traffic types. We summarize this development in Table 3.1.

All the networks in this table, except Frame Relay and SMDS, implement layers 1 and 2 of the OSI model. In Chapter 4 we describe the Internet Protocol or IP, which roughly correspond to layers 3 and above of the OSI model. IP can be implemented on top of layer 2 of these networks. Frame Relay and SMDS

Name	Speed, connectivity	Delay	Application
Ethernet	10-100-1000 Mbps, local area	Random, increases with load	Transfer of messages between nearby computers
Token ring	4, 16 Mbps, local area	Random but bounded	Transfer of messages between nearby computers, some real- time traffic
FDDI	100 Mbps, LAN and campus	Random but bounded	LAN interconnections, real-time and CBR applications
DQDB	Unspecified	Random	Unspecified but similar to FDDI
Frame Relay	1.5 Mbps, wide area	Random, increases with load	Transfer of messages between distant computers
SMDS	1.5 to more than 45 Mbps	Random, traffic shaping	LAN interconnections, migration to ATM

**3.1**

Summary of advances in packet-switched networks.

**TABLE**

also implement layer 3 (the network layer). By means of a router with some additional functionality, these networks can also implement IP.

Over its 100-year history, the services provided by the telephone network seem hardly to have changed. The quality of voice is much improved, connections can be set up through "direct dialing" to virtually anywhere in the world, service is very reliable, and costs have steadily declined. Nonetheless, users must find these improvements slow, cumulative, and unremarkable. To appreciate the progress of the telephone system, one has to study the changes in its infrastructure: the increased throughput and capabilities of its switches and its links.

The 20-year history of packet-switched networks presents a sharp contrast. From the viewpoint of user applications, its progress has been dramatic. Starting from the humble beginnings of interconnecting a computer with a printer or another computer, these networks have expanded to enable users around the world to communicate in the form of data, text, images, movies, and animation.

Several technical advances cooperated in bringing about this dramatic progress. The acceptance of the principles of layered architectures for de-

veloping new services encouraged modularity and reuse of existing services and permitted network engineers and application developers to work independently. The choice of transporting packets as the basic service turned out to be ideally suited for interconnecting computers and other devices, because it isolated the rapid technical advances in link speed from the advances in software. Finally, the invention of local area networks such as Ethernet quickly reduced the cost of access at the same time as their speed increased, and the capabilities of workstations and personal computers kept up with the improvements in access and speed.

---

### 3.9

### NOTES

The OSI model is described in detail in [T88, W98]. Standards for local area networks such as Ethernet, token ring, and FDDI are published by the IEEE. Fast Ethernet is described in [COUCR97, MW96], gigabit Ethernet is described in [F98]. One wireless LAN product with a CSMA MAC protocol is described in [CMM94]. Ethernet switches and VLANs are discussed in [MW96] and in technical documentation of switch manufacturers.

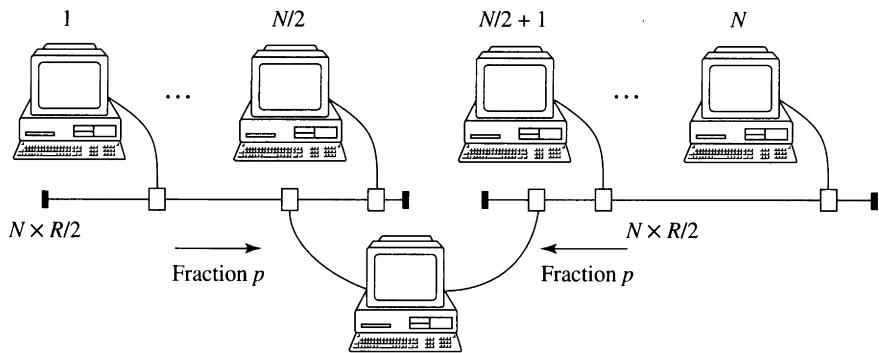
Frame Relay, SMDS, and SONET (discussed in Chapter 5) are described in [B95]. Frame Relay and ATM interworking efforts are reviewed in [DE96].

---

### 3.10

### PROBLEMS

1. There are many cases in which a link is shared by different devices. In such cases there is some explicit or implicit protocol that regulates the access to the link. Discuss the following examples.
  - (a) Computer backplane bus is used by many devices (e.g., CPU, memory, I/O devices) to communicate. How is access to this common bus regulated?
  - (b) In a cellular phone system, a fixed number of channels are available for use by all the mobile phones within the same cell. (The ratio of number of phones to the number of channels is the “pair gain.”) How does a mobile phone user get access to an idle channel or learn that no channel is idle?
  - (c) There are a fixed number of seats in a theater or bus or airplane. Many users may wish to occupy a seat. How is access regulated? Airlines



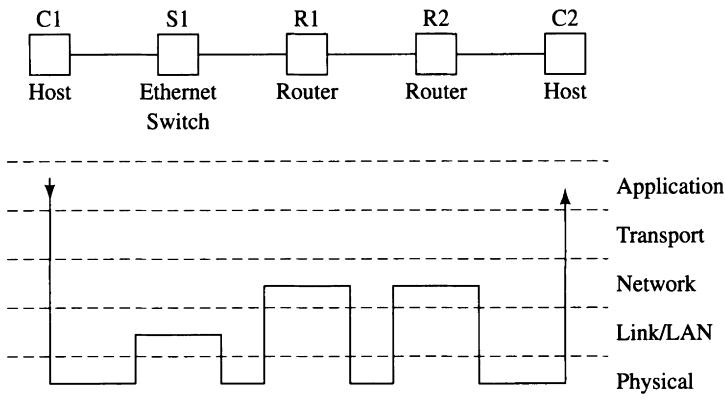
3.35

FIGURE

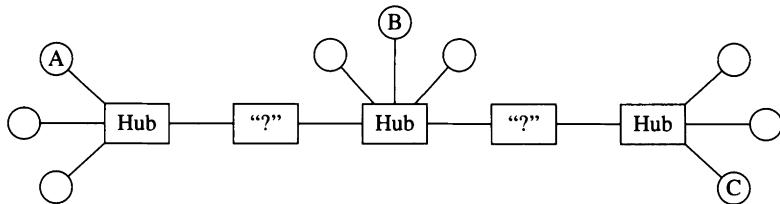
By partitioning an Ethernet into work clusters, the network manager can accommodate more nodes.

sell more tickets than seats because people cancel their flight. How is access regulated when there is “overbooking”?

- (d) When a driver wishes to change into an adjacent lane in which there are other cars, the lane-change maneuver needs to be coordinated to prevent an accident. What “protocol” do drivers use to achieve coordination?
2. Figure 3.35 shows how network managers can partition an Ethernet network to accommodate more nodes. The figure assumes that each node transmits  $R$  bps, on average, during a representative period of time. There are  $N$  nodes to be connected. If the total transmission rate  $N \times R$  is larger than the rate that can be handled by one Ethernet, then the network manager can try to partition the network. For instance, if the efficiency of one Ethernet connecting the  $N$  nodes is 80% and if  $N \times R > 8$  Mbps, then one Ethernet cannot handle all the nodes. Let us assume that the  $N$  nodes can be divided into two groups that do not exchange messages frequently. For simplicity, say that each group sends a fraction  $p$  of its messages to the other group. Let us connect the computers in each group with a dedicated Ethernet, as shown in the figure. The two Ethernets are connected by a bridge or by a device called a *switching hub* that performs the same function between Ethernet segments. Calculate the traffic on each Ethernet. The arrangement can handle the  $N$  nodes if this rate is less than the efficiency of each Ethernet times 10 Mbps. Of course, the bridge must be able to handle the traffic.
3. Discuss the flow of information at the correct layers in a connection from C1 to C2.



4. (a) Consider the network topology depicted in the figure below where the boxes labeled with "?" are Ethernet hubs.
- (1) Redraw (roughly) the figure and indicate the extent of the Ethernet "collision domains."
  - (2) Does node A's ARP table ever contain entries for node B and node C network addresses?
  - (3) When node A is sending a packet to node C, what Ethernet address does node A use for the destination?
  - (4) Suppose there is two-way communication between nodes A and B. Does node C get to see this traffic?
- (b) Assume that the boxes labeled with "?" are Ethernet switches. Answer the same sub-questions from (a).
- (c) Assume that the boxes labeled with "?" are Internet Protocol routers. Answer the same sub-questions from (a).

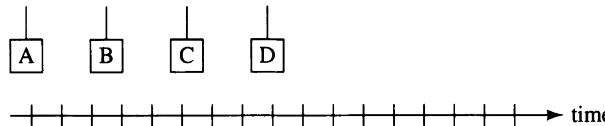


5. The OSI transport layer segments a message into packets and appends a sequence number to each packet so that the receiver can reassemble the original message. In practice, the sequence number is selected modulo  $N$ , for some integer  $N$ , and consecutive packets are numbered  $0, 1, \dots, N - 1, 0, \dots$ . (For example, in SMDS the sequence number has 4 bits, so  $N = 16$ .) With this choice of sequence numbers, the receiver would be

unable to detect the loss of  $N$  consecutive packets of the same message. How large should  $N$  be to keep the probability of missed detection small? How large is  $N$  for IP, ATM, SMDS?

6. In this problem you design a protocol between two nodes that recovers when one of the nodes crashes. Node A sets up a connection with node B. Node B sends packets to A using GBN. Node B closes the connection when it has sent all the packets. Make sensible assumptions about the channels between the two nodes.
7. Modify Figure 3.21 to reflect the release after transmission protocol used by the 16-Mbps token ring network. Show that its efficiency is approximately  $(1 + a/N)^{-1}$ , which approaches 100% as  $N$  becomes very large.
8. The physical layer of a *token bus network* is a broadcast coax cable like Ethernet, but the MAC layer is like the token ring. Each station on the bus has a number 1, 2, . . . , say. After station  $i$  transmits its packets, it releases a token, indicating that the token is intended for station  $i + 1$ . When the last station, say  $N$ , finishes transmission, it releases the token, indicating that the next station is station 1. Analyze the efficiency of the token bus network.
9. Suppose there are  $N$  stations on an Ethernet. Suppose the probability is  $p$  that any of them has a packet ready for transmission and suppose that different stations are independent. What is the probability  $p_m$  that  $m$  stations have a packet ready for transmission,  $m = 0, 1, \dots, N$ ? The average number of packets ready for transmission is  $\lambda := pN$ . Suppose that  $N \rightarrow \infty, p \rightarrow 0$  in such a way that  $N \times p \rightarrow \lambda$ . For this asymptotic case, show that the probability that there are  $m$  packets to transmit is Poisson with mean  $\lambda$ .
10. Consider the wireless packet network shown in the figure on the next page. Four stations share a slotted ALOHA channel by transmitting with probability 0.25 each in every slot, independently of one another. In addition to the risk of collision, packets that are transmitted also face transmission errors. Assume that errors corrupt packets that were otherwise successfully transmitted with probability 0.1. Acknowledgments are given priority and are corrupted by errors with probability 0.1.

Specifically, assume that A transmits a packet to B. If the transmission is successful (possibly with errors), then B gets the channel for one slot to transmit one ACK: no other station interrupts the transmission of B's ACK. With this procedure, the transmission of a packet by A is complete with some probability after 2 slots. If that transmission is not complete, then A knows that it must try to transmit that packet later.

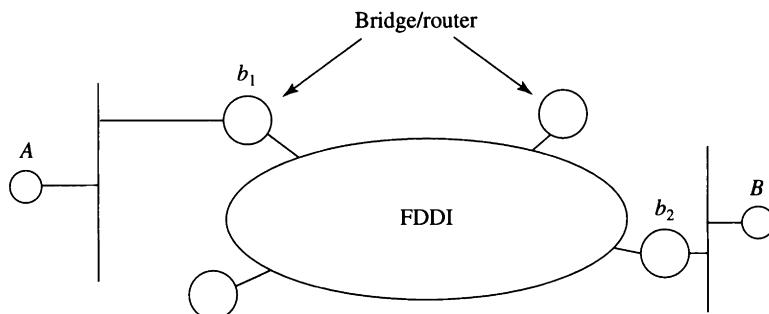


Calculate the average reliable throughput of packets over this channel, in packets per slot.

11. If your computer is connected to an Ethernet, find out what the network utilization is, how many collisions have occurred, and other network statistics that are available. What is the maximum sustained rate at which your computer can generate network traffic? What is limiting this rate? Is it the disk, I/O bus, or some other factor?
12. Recall the FDDI MAC protocol.
  - (a) Show that the maximum time between successive token arrivals in the FDDI network is  $\text{TTRT} + \text{TRANS}$ .
  - (b) Show that this time is  $2 \text{ TTTRT}$  (neglecting packet transmission time) when synchronous traffic is included.
  - (c) Show that when provision for synchronous traffic is made, the token arrivals are not periodic. What is the maximum deviation in the token interarrival time for synchronous traffic? How much buffering would be needed to transfer constant bit rate synchronous traffic?
  - (d) Give a definition of fairness and argue that FDDI provides fair access to all stations.
  - (e) The DQDB MAC protocol attempts to regulate access as if all the stations placed their packets in a single queue that is served on a first-come, first-served (FCFS) basis. Would you call FCFS access fair? Why?
13. We want to analyze the efficiency of DQDB similar to the analysis of the token ring network in section 3.3.2. Suppose there are  $N$  equally spaced stations, indexed  $1, 2, \dots, N$ , arranged from left to right, and suppose  $\text{PROP}$  is the propagation time between adjacent stations. Suppose  $T$  is the transmission time at each station of a 53-byte frame. (For example, if the link speed is 100 Mbps, then  $T = 53 \times 8 \times 10^{-8}$  s.)
  - (a) Suppose the head station transmits  $F$  idle frames per second, back to back. What is  $F$  (in terms of  $T$ )?
  - (b) We define utilization as the fraction of frames that carry data. Suppose that at all times at least one station has some information to send to its right. Show that the utilization of bus A in Figure 3.27 is 100%.
  - (c) Consider two stations  $i$  and  $j$  with  $i$  to the left of  $j$ . Suppose both are transmitting to a station  $k$  on their right. Suppose  $i$  and  $j$  receive packets

to transmit at times  $t_i$  and  $t_j$ . Suppose  $t_i > t_j$ , that is,  $i$ 's packet arrived later than  $j$ 's packet. Show that the DQDB protocol works in such a way that  $j$  will transmit its packet before  $i$ . Now suppose  $t_i < t_j$ . Construct an example such that the protocol will transmit  $i$ 's packet after  $j$ 's. What is the maximum amount of time  $t_j - t_i$  for which this could happen? (Note: In a true first-come, first-served system,  $i$  should transmit before  $j$  whenever  $t_i < t_j$ .)

14. Consider the expressions shown in Figure 3.28. Show that if transmission from each node to the next corrupts a packet with probability  $1 - p$ , and if each frame is retransmitted until it is successfully received, then each frame is transmitted  $1/p$  times on average before being successfully transmitted. Calculate the packet error rate  $1 - p$  for packets of size 100 and 1,000 bytes and bit error rates of  $10^{-4}$  and  $10^{-8}$ .
15. Suppose in Figure 3.28 that  $\sigma_1 = 50$  ms and  $\sigma_2 = 3$  ms. Suppose  $k = 10$ , so there are 10 nodes. How small should  $(1 - p)$ , the packet error rate per link, be before Frame Relay has less delay than X.25?
16. Suppose an FDDI network connects two Ethernets as shown in Figure 3.36. Bridge  $b_1$  is attached to one Ethernet and the FDDI, while  $b_2$  is attached to the other Ethernet and the FDDI. These bridges are supposed to provide transparent routing. Explain how computer  $A$  can send packets to  $B$  using  $B$ 's MAC address? Explain how the bridges can obtain their address conversion tables.



3.36

**FIGURE**

Bridges  $b_1$  and  $b_2$  must be designed so that computer  $A$  can transparently send packets to  $B$  connected to a different Ethernet. Note that the MAC addresses on FDDI and Ethernet are different.

# The Internet and TCP/IP Networks

This chapter explains the Internet and networks that use the same protocols. We call these networks *TCP/IP Networks* because TCP and IP are their most important protocols. For instance, companies build isolated TCP/IP networks called *intranets* to connect their computers and servers. In this chapter we first discuss the Internet and its topology. We then describe the IP and TCP protocols, and some of the major applications, including the http protocol used in the World Wide Web. We also discuss performance characteristics of the TCP/IP networks and proposed upgrades of the IP and TCP protocols.

Section 4.1 explains the Internet. Section 4.2 clarifies the addressing and routing and reviews the layered structure of the protocols. Section 4.3 discusses the Internet Protocol. Section 4.4 describes the TCP and UDP protocols and considers important applications. Section 4.5 looks at both the success and the limitations of the Internet. Section 4.6 discusses performance issues of the Internet.

---

## 4.1

## THE INTERNET

The Internet today interconnects a large number of computers and networks throughout the world. There were 1 million such computers in early 1993, 5 million in 1995, 16 million in 1997, and over 50 million in 1999 organized in 2 million domains.

The Internet has its origin in the ARPANET network sponsored by the U.S. Department of Defense starting in the 1960s. The ARPANET was a datagram store-and-forward network that the Department of Defense liked for its ability to reroute packets around failures. This feature makes datagram networks survivable. Another important objective of ARPANET was to enable the interconnection of heterogeneous networks. The technical success of the Internet is due to the large variety of applications (from e-mail and telnet, to file transfer and WWW) that IP can support on the one hand and, on the other hand, the many different networks that can implement IP. (See Figure 4.2 in section 4.2.) We will discuss this key feature at the end of this section.

A major factor that contributed to the popularity of the Internet was the exploitation of network externalities. This was achieved through early standardization and free distribution of its protocols and their software implementations, which could run on personal computers as well as on workstations and mainframe computers. Network externalities were created by the National Science Foundation's subsidy of the construction and use of the Internet. Since 1995, expansion and development have been funded by private enterprise.

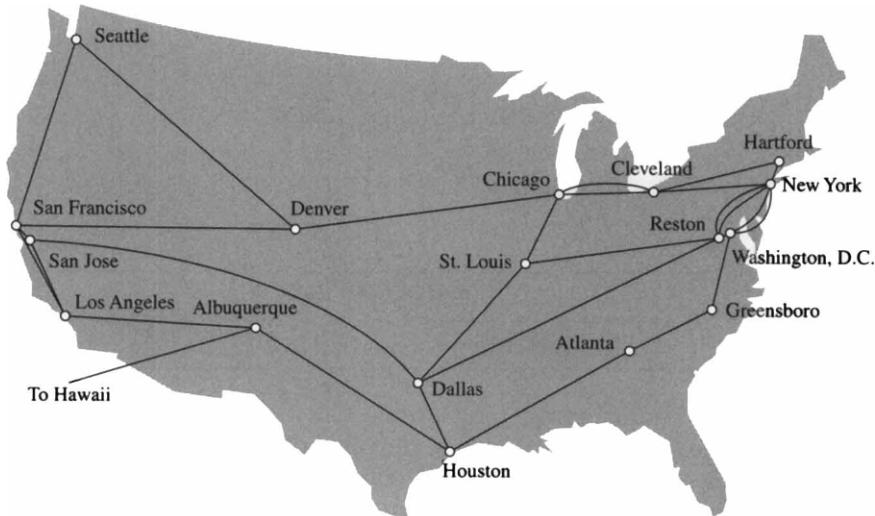
The spectacular growth of the Internet is fueled largely by the World Wide Web, which makes multimedia information available at the click of a mouse. The vast increase in demand for Internet traffic explains the rapid growth in network capacity. It is estimated that in 1999 the volume of data traffic is comparable to that of voice. Moreover, because data traffic doubles every year whereas voice traffic increases only by about 10% a year, voice traffic will soon amount to only a small fraction of the total. These observations lead to the conclusion that the future network should be optimized for data and, in the background, should be able to carry voice traffic reliably and with a small delay.

The Internet is a network of networks. It comprises backbone networks of point-to-point links that connect regional gateways called *network access points* (NAPs). Routers attached to the NAPs are called *points of presence* (PoPs). Subscribers connect to a PoP either with a dial-in modem over their telephone line, or with a digital subscriber line (typically ADSL), a cable modem, or a leased digital line. Some businesses connect to a PoP with an optical link. The customer's computers are typically interconnected with a local network.

Over time, and incrementally, link speeds have increased from 56 Kbps to 1.5 Mbps to 45 Mbps. The recent explosive growth in demand is being met by 155-Mbps, 622-Mbps, and higher-speed (2.4 Gbps and 9.6 Gbps) SONET links,

leased from telephone companies. Local area network speeds have increased as well to 100-Mbps Ethernet LANs and FDDI rings, Gbps Ethernet LANs, and soon to 10-Gbps Ethernet LANs. The Internet is used for applications that require (relatively) low transmission rates and that can tolerate large delays. Recent advances are aimed at accommodating applications that need higher network performance.

Until April 1995, the backbone of the Internet was managed under the auspices of the National Science Foundation. The topology of the backbone was simple, with four public NAPs and two private ones. When competing commercial carriers took over the backbone, the topology and routes became much more complex. Today, there are 75 public NAPs around the world, 12 of them in the United States. Many telecommunication companies are building U.S.-wide Internet backbones. As of May 1999, these backbones include those of ANS, AT&T, BBN/GTE, CERFNET, DIGEX, EBONE, MCI, NETCOM, PSI, Qwest, Sprint, UUNET, and Verio. Figure 4.1 shows the topology of the ANS backbone. ANS (Advanced Networks & Services, Inc.) was a subsidiary of America On Line until 1998 when it was acquired by WorldCom. You can find the maps of the other backbones at <http://boardwatch.internet.com/isp>. Much of the traffic is routed through private interconnections. These "private



4.1  
FIGURE

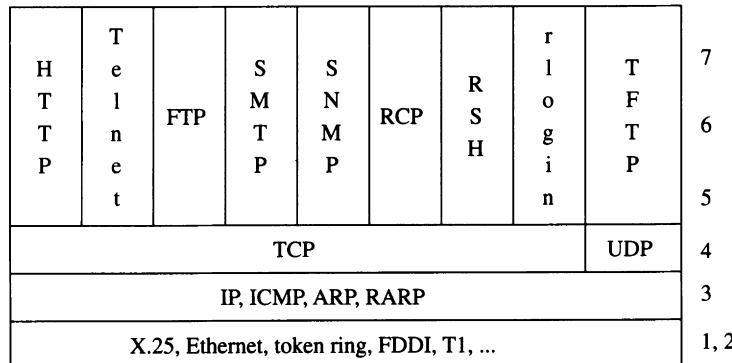
ANS backbone. This is one of many U.S.-wide backbones deployed by various telecommunication companies.

"peering" arrangements between ISPs can take place anywhere the transaction is mutually convenient, and they account for an estimated two-thirds of all Internet traffic.

## 4.2 OVERVIEW OF INTERNET PROTOCOLS

In this section we describe the main components of the Internet protocols. The Internet protocol layers are shown in Figure 4.2. The figure shows the correspondence between the IP protocols and the OSI seven-layer model. The correspondence is inexact. Many networks implement layers 1 and 2, including those studied earlier. Layer 3, the network layer, is implemented by IP. Layer 4, the transport layer, is implemented by two protocols, UDP and TCP. There is no direct counterpart to the higher OSI layers.

As Figure 4.2 shows, the Internet protocols are *internetworking* protocols. That is, these protocols are designed to glue together many different networks such as Ethernet LANs, FDDI, wireless networks, and point-to-point links. Each



## 4.2

### FIGURE

Internet protocols are arranged in a layered hierarchy and compared with the OSI seven-layer model. On top of the basic IP or network layer and TCP/UDP or transport layer are generic applications such as Web page transfer (Hypertext Transfer Protocol or HTTP), file transfer (File Transfer Protocol or FTP), remote file copy (rcp), remote terminal (Telnet), remote login (rlogin), network management (Simple Network Management Protocol or SNMP), and e-mail (Simple Mail Transfer Protocol or SMTP). User applications such as WWW and Microsoft's Outlook are built on top of these generic applications.

of these networks uses its specific addressing conventions, packet format, and protocols. The IP protocol sees these networks as implementing virtual links between their nodes and does not concern itself with the characteristics of these links. The only assumptions that the IP protocol makes about these virtual links is that they can transport packets of up to some maximum size. The virtual links may be unreliable and have a wide range of transmission rates and delays.

The IP protocol uses these virtual links that the individual networks implement to deliver packets end-to-end across a number of such links. This end-to-end delivery necessitates two basic tasks: addressing and routing. The IP protocol defines globally unique addresses for the host computers and routers. These addresses are independent of those already defined by the specific networks to which the hosts are attached. That is, if a computer is on an Ethernet LAN, the Ethernet interface card of that computer has a 48-bit MAC address. If this Ethernet LAN is attached to a TCP/IP network, then the Ethernet interface of the computer also has at least one IP address. The IP addresses are organized in a hierarchical way. In this organization, the computers are grouped into clusters called *IP subnets*. From the IP address of a computer, one can determine its IP subnet. The IP routing uses this hierarchical structure of the addresses to reduce the size of the routing tables that the routers maintain.

We explained that the Internet Protocol delivers packets from end to end in the Internet or, more generally, in a TCP/IP network. To make these packet deliveries more directly usable by applications, the protocols of layer 4 perform a few additional tasks: multiplexing, error control and reordering, flow control, and congestion control. *Multiplexing* means adding a “port number” to the packets to distinguish packets destined to different applications. An application “transmits” and “listens” to a specific port. *Error control* means arranging for packets that do not arrive correctly to be retransmitted; *reordering* means delivering the packets in their order of transmission. *Flow control* means stopping the transmissions when the destination runs out of buffer space. Finally, *congestion control* means slowing down when packets are getting dropped in the network, which indicates that the network is probably congested. We explain these additional tasks in the following pages. For now, we note that UDP, the user datagram protocol, only adds multiplexing to the packet delivery and discards corrupted packets. TCP, the transmission control protocol, adds all the tasks that we mentioned so that it implements error-free, ordered, and well-paced deliveries of packets.

In the sections that follow, we examine IP, UDP, and TCP. We also discuss HTTP, TFTP, and a few other application layer protocols in some detail.

## 4.3 INTERNET PROTOCOL

The network layer of Internet, called the *Internet Protocol* (IP), is the most important. The version in use, first implemented in 1984, is IPv4. (See RFC 791.) Version 5 was used for some experiments. Version 6, IPv6, is rarely implemented in 1999 and may be implemented more widely over the next few years.

As we explained in the previous section, IP organizes the addresses hierarchically and maintains the routing tables of the routers. In addition, IP reports some delivery problems.

We start by describing IPv4, the multicast and mobile version of IP, then we explain the modifications of IPv6.

### 4.3.1

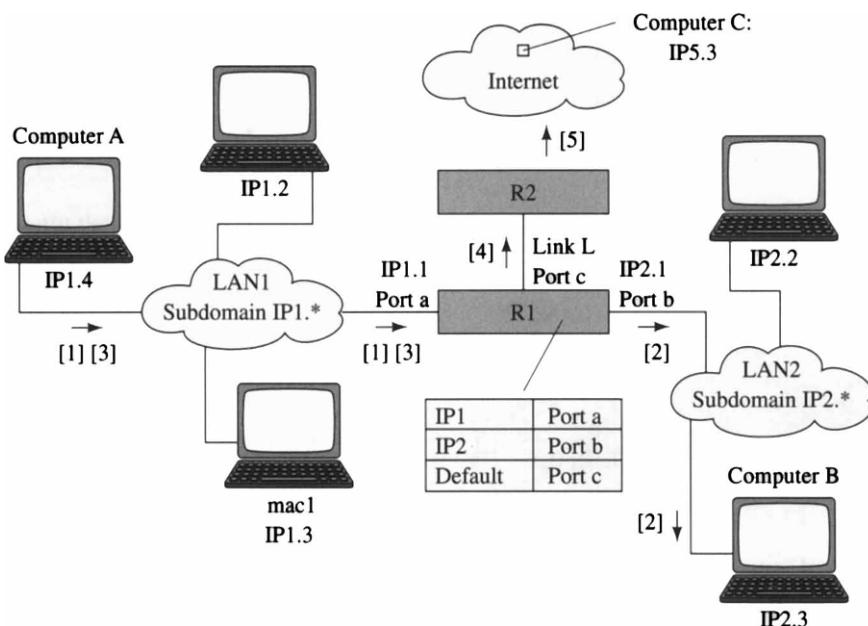
### IPv4

In terms of the Open Data Network model of section 2.8, the IP bearer service is the delivery of messages in the form of datagrams of size up to  $2^{16}$  bytes (64 Kbytes). This delivery service carries no guarantee of service quality in terms of error, delay, or bandwidth. Such service is called *best-effort* service, meaning thereby that the network will attempt to do the best it can. IP discards packets whose header is corrupted.

Figure 4.3 illustrates the main ideas of IP routing. The figure shows two LANs that are attached together with router R1 and to the rest of the Internet with router R2 at the PoP of the ISP. The link between R1 and R2 uses some specific framing that we do not examine here. Each computer has a LAN address and an IP address. The IP addresses of the computers of LAN1 have the form IP1.*x* and those of LAN2 have the form IP2.*y*. The router R1 maintains the routing table shown in the figure. This table specifies where R1 should send the packets next.

Assume that computer A with IP address IP1.4 wants to send [ data 1 ] to computer B with IP address IP2.3. Here are the steps that are required for this packet transfer:

1. Given the name of computer B, computer A discovers its IP address IP2.3, by using a directory service called DNS.
2. Computer A places [ data 1 ] in an IP packet with source address IP1.4 and destination address IP2.3. This IP packet is [ IP1.4 | IP2.3 | data 1 ].
3. Computer A determines that it must send [ IP1.4 | IP2.3 | data 1 ] to R1. To make this determination, computer A notes that the IP address IP2.3 is not



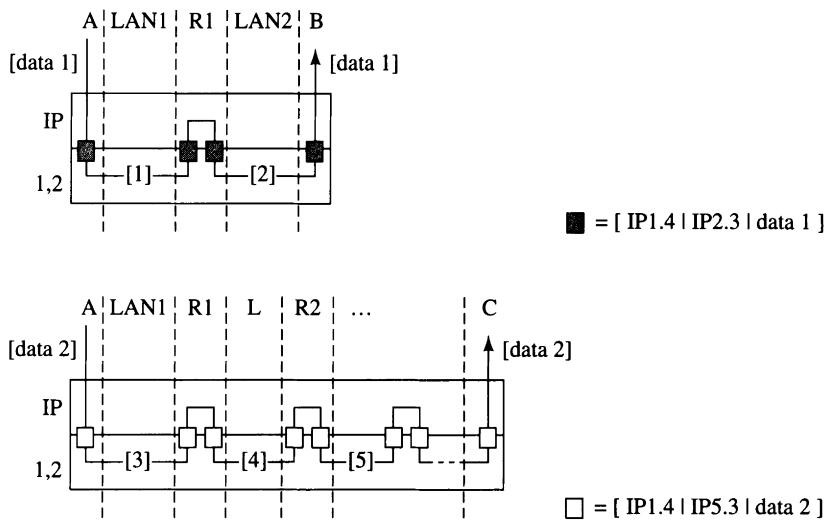
## 4.3

**FIGURE**

A TCP/IP network. The figure shows two LANs attached to the rest of the Internet.

on LAN1 as it is not of the form IP1.x. Computer A is configured with the address IP1.4 of the “default gateway” R1 to which it must send packets that leave LAN1.

4. To send [ IP1.4 | IP2.3 | data 1 ] to R1 over LAN1, computer A places it in a frame of the format required by LAN1. For instance, if LAN1 is an Ethernet LAN, then that format looks like [ mac(IP1.1) | mac(IP1.4) | IP1.4 | IP2.3 | data 1 | CRC ], where mac(IP1.1) and mac(IP1.4) are the MAC addresses of the network interfaces of R1 and A on LAN1, and CRC is the error-detection field. We designate that frame by [1] in the figure.
5. When it gets the packet, R1 removes it from its Ethernet frame and recovers [ IP1.4 | IP2.3 | data 1 ]. R1 then consults its routing table and finds that the subnet with addresses IP2.y is attached to port b.
6. To send [ IP1.4 | IP2.3 | data 1 ] to computer B over LAN2, R1 places it into a frame with the format suitable for LAN2. We designate that frame by [2] in the figure.



4.4

FIGURE

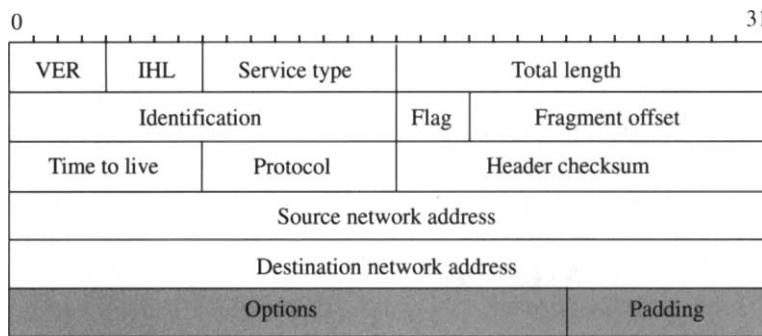
Symbolic representation of the transfers of packets at layers 1 through 3.

7. Eventually, computer B gets the packet, removes it from its LAN2 frame and from its IP envelope and extracts [ data 1 ].

The top part of Figure 4.4 shows the packet transfers at the different layers. The packet [ data 1 ] arrives at the IP layer in computer A. The IP layer adds the IP addresses of the source and destination. The IP layer also finds out that the packet must go to R1. The IP layer then gives this IP packet to the LAN1 layer that delivers it to R1. This transfer is in a format [1] suitable for LAN1. When it gets the packet, R1 invokes its IP layer, which determines where to send the packet next. The IP layer gives the packet to the LAN2 layer, which delivers it to computer B.

Note from this diagram that IP views LAN1 and LAN2 as virtual links. Note also that the LANs are not aware of the IP addresses.

Figure 4.3 also shows the transfer of [ data 2 ] from computer A to computer C with IP address IP5.3 somewhere on the Internet. The steps are similar, except for step 5. When it gets the packet [ IP1.4 | IP5.3 | data 1 ], R1 consults its routing table. Since no row corresponds to addresses of the form IP5.z, R1 sends the packet on the “default” port, port c. Figure 4.4 shows the transfers at the various layers. We designate by [3] the packet [ IP1.4 | IP5.3 | data 1 ] in the format for LAN1 from A to R1, by [4] that packet in the format for link L from



**4.5**  
**FIGURE**

The IP header specifies the source and destination addresses and control fields described in the text. (VER = version; IHL = header length/32 bits.)

R1 to R2, and by [5] that packet in the format for the link that exits R2 toward the rest of the Internet.

Each IP packet has an IP header of at least 20 bytes. (See Figure 4.5.) The header indicates the source and destination network addresses of the message. The IP header also specifies the time to live of the packet. When a router gets an IP message, it decrements its time to live and it discards the message if the time to live reaches 0. This procedure prevents packets from floating around for a long time in the network in case of routing errors. The header indicates the type of service or TOS requested by the message: either low delay, high throughput, or very reliable. TOS is ignored in current implementations, although it could be used for Differentiated Services (RFC 2474) and also for Explicit Congestion Notification (RFC 2481).

The IP header also specifies the upper-level protocol (typically UDP or TCP) that requested the transmission of the message so that the receiver knows how to handle the message. Finally, the header may request some optional services such as recording the route followed by the packet (the list of nodes is then written in the IP header as the packet progresses), following a route specified by the source, or time-stamping of the message by each node that it goes through.

We explain three important aspects of IP: addressing, fragmentation/reassembly, and routing.

**Addressing** There are four important addresses: MAC (layer 2) or hardware addresses, IP or network (layer 3) addresses, domain-based addresses used, for example, in e-mail, and Universal Reference Locator or URL.

Layer 2 addresses vary from one network type to another: Ethernet and Token Ring use 48-bit addresses, Frame Relay uses a 12-bit DLCI connection identifier, and SMDS uses telephone numbers. In order for the millions of computers on different networks on the Internet to communicate, they must use the unique Internet or IP address assigned to them. Strictly speaking, the IP address is assigned to a host's network interface. A computer, such as a router, with more than one network interface, has a different IP address for each interface. (A computer's layer 2 interface to an Ethernet, say, will also have a MAC address.)

To facilitate routing of packets, the 4-byte *IP address* (written as a set of four decimal numbers separated by a dot [ . ]) is divided into a two-part hierarchy: the first part is the network number, the second part is the host number. A packet is routed, hop by hop, from one network router to an adjacent router until it reaches the destination network whose router forwards it to the destination host. Thus a computer with IP address 123.32.239.151 has the 16-bit network number 123.32 and the 16-bit host number 239.151.

The original organization of IP addresses was class-based. Although the current usage of addresses is now classless (based on subnets), we explain the class-based addressing to motivate the current structure and also because it is still occasionally used. In class-based addressing, there are five classes of addresses. Class A addresses have 8-bit network numbers beginning with 0 and 24-bit host numbers, so one class A network may have up to  $2^{24}$  hosts. Class B addresses have 16-bit network numbers beginning with the bit pattern 10 and 16-bit host numbers, so one class B network may have  $2^{16}$  hosts. Class C addresses have 24-bit network numbers beginning with the pattern 110 and 8-bit host numbers, so one class C network can have 256 hosts. Class D addresses, beginning with 1110, for multicast, are discussed on p. 173. Class E addresses, beginning with 1111, are reserved for future use.

The class B network 123.32 can have  $2^{16} = 65,536$  hosts. A router in this network should be able to forward a packet addressed to any of those hosts. This can lead to very large routing tables. (Routing tables are described below.)

*Subnetting* simplifies the management of addresses by partitioning the address space so as to define a single network that includes many different physical segments. Subnetting groups all the IP addresses with the same leading  $n$ -bits into one network. Each IP address is accompanied by a 32-bit *subnet mask* that consists of  $\frac{n}{16}$  ones followed by  $32 - n$  zeros. The subnet mask is written in the same notation as IP addresses. For instance, 255.255.255.0 corresponds to  $n = 24$ . In the example of the computer with IP address 123.32.239.151, one can define a subnet that consists of all the IP addresses that start with 123.32.239. This subnet is a subset of the class B network 123.32. With this subnet defini-

tion, host A belongs to the subnet 132.32.139 and it has number 151. A router within network 123.32, when it receives the packet for host A, would have the simpler task of determining whether A belongs to the router's own subnet. The router does this by taking the logical AND of A's IP address and the router's subnet mask 255.255.255.0, which yields A's subnet number 123.32.239. If this matches the router's subnet number, then A is a local address; otherwise the router looks up its routing table to determine the address of the router for A's subnet. Typically, subnet numbers correspond to a LAN, in which case "local address" means an address on the router's LAN. (For example, 123.32.239 is the subnet number of an Ethernet LAN.) A change of addresses inside the subnet does not affect the routing tables of the routers.

*Classless interdomain routing* or CIDR (RFC 1519) extends the idea of subnets to variable-length subnet masks. CIDR uses the address space efficiently by reserving only the number of addresses that are needed for a subnet (approximately). In CIDR, a group of hosts whose addresses have a common prefix are grouped as a subnet. This common prefix serves as a network number for the group as a whole. For example, the 1024 addresses in the four class C networks 192.0.8.\* , 192.0.9.\* , 192.0.10.\* and 192.0.11.\* , have the common prefix 192.0.8 with a subnet mask of 255.255.252.0. Another subnet might correspond to the prefix 192.0.2 and yet another to the prefix 192.0.5. The routing decision, to identify the correct subnet, is a longest-matching-prefix search. For instance, if the destination address is 192.0.9.123, the router must find out that it belongs to the subnet with prefix 192.0.8.

The pool of IP addresses allocated to an organization may be insufficient to assign an individual IP address to every host. Besides, only a few hosts may connect to the Internet at any time, so that the pool can be shared, with unused IP addresses assigned on a temporary basis. (An example is an Internet Service Provider [ISP], which has many subscribers, only a few of whom are logged on at any time.) The *Dynamic Host Configuration Protocol* or DHCP is used for this purpose (RFC 2131). A host, needing an IP address, uses its MAC address to broadcast a *DHCP discover* packet. DHCP servers reply with a *DHCP offer* that includes an unused IP address, and the host accepts one of those offers and broadcasts its selection with a *DHCP request*. The designated server commits its offer with a *DHCP ack*. The address has a time to live and must be refreshed to remain valid. When the host is done, it can send a *DHCP release*; otherwise the address is automatically released when its lifetime expires.

A distributed directory service called the *Domain Name System* or DNS is provided by the Internet to translate between IP addresses and names, and to control Internet e-mail delivery. The names are hierarchically organized,

administered in a decentralized manner, and the directory service is provided by distributed DNS servers.

Domain names are organized in a tree. The top-level domains are *edu* (educational institutions), *com* (commercial), *gov* (U.S. federal government), *org* (nonprofit organizations), *net* (mostly network providers), *int* (some international organizations), *mil* (U.S. armed forces), and the country-specific domain (e.g., *fr*, *gr*, *jp*, etc.). The *edu* domain server has pointers to *berkeley.edu* name servers, and others like it. Thus a domain is a subtree of this domain name system. The *berkeley.edu* domain includes subdomains like *eecs.berkeley.edu*, which includes host names like *diva.eecs.berkeley.edu*. The registration authority for *berkeley.edu* is allowed to assign subdomain names like *eecs.berkeley.edu* and may delegate authority to assign names under it, like *diva.berkeley.eecs.edu*.

The authors use a computer named *diva.eecs.berkeley.edu*. This name consists of the institution name (*berkeley.edu*) and the name of the computer (*diva.eecs*). To send a message to one of the authors, you send it to *varaiya@diva.eecs.berkeley.edu* or to *wlr@diva.eecs.berkeley.edu*, respectively. You do not need to know where *diva.eecs.berkeley.edu* is located. The directory service translates the name into the address 128.32.110.56, and the network layer finds a path that leads to that computer.

The directory servers supervise disjoint zones of the name-structure tree. For reliability, multiple servers are used for each zone. For the purpose of illustrating how DNS works, we assume that one zone represents the children of the *edu* node, another represents the children of the *berkeley.edu* node, and another represents the children of the *eecs.berkeley.edu* node. Note that the zones do not have to be all the children of one node but can be defined more generally.

Assume that you want to send a packet to *diva.eecs.berkeley.edu*. Your local directory service sends the request “find the IP address of *diva.eecs.berkeley.edu*” to the *edu* zone name server. This server knows the address of the *berkeley.edu* zone server and forwards the request to that server. This server knows that *eecs* has its own server, and it forwards the request to that server. Finally, the *eecs* server returns the address of *diva* to the *berkeley.edu* server, which returns it to the *edu* server, which finally sends it to your computer. The servers cache the intermediate answers. For instance, the *edu* server caches the address of the *berkeley.edu* server. If a requested address is already cached in a server, then it replies to the query without having to forward the request. Cache entries have a time to live to prevent obsolescence.

The Internet Corporation for Assigned Names and Numbers (ICANN, [www.icann.org](http://www.icann.org)) was formed in 1998 to take over responsibility for IP address space allocation, protocol parameter assignment, domain name system management, and root server system management.

Whereas MAC and IP addresses refer to a host interface, the *Universal Reference Locator* or URL specifies the location of resources (such as files, directories, html documents, or database services) and a scheme for retrieving it via the Internet. A URL is written as

```
<scheme> :<scheme-specific-part>
```

The common schemes are ftp, http, gopher, news, mailto. The scheme-specific-part is of the form

```
//<user>:<password>@<host>:<port>/<url-path>
```

User, password, and url-path may be omitted, and the port number may be assigned by default. For example `ftp://ftp.cis.ohio-state.edu/pub/` is a URL of the anonymous FTP archive of the Computer and Information Science Department of Ohio State University. Files located in the archive are accessed by the File Transfer Protocol (FTP). The default port for FTP is 21.

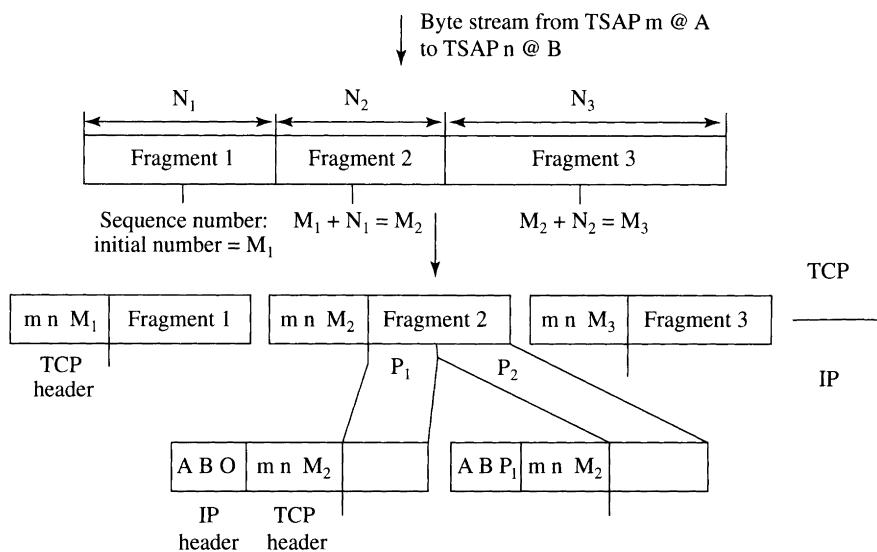
An HTTP URL takes the form

```
http://<host>:<port>/<path><searchpart>
```

where host and port are as above, path is an HTTP selector, and searchpart is a query. The default port for HTTP is 80.

**Fragmentation/Reassembly** The OSI model specifies that layer 4 (the transport layer) perform the fragmentation and reassembly of messages. However, in a TCP/IP network, IP fragments messages into IP packets in the source for the data link layer and reassembles IP packets into messages at the destination. For instance, if the messages go through an Ethernet, then they must be divided into packets of at most 1.5 KB. A router may have to further fragment IP packets that are larger than the maximum transfer unit of the next link. In any case, only the destination reassembles the messages. The header contains an identification number of the message to which the packet belongs and an offset indication that specifies the position of the data portion of the IP packet inside the original message. The 3-bit flag indicates whether the router may fragment the packet or not and whether there are additional fragments or the packet is the last fragment of the message. The fragmentation/reassembly function is shown in Figure 4.6.

**Routing** To route datagrams, Internet nodes called *routers* maintain routing tables. The table specifies where the node should send a datagram next. If the datagram is destined to a network to which the router is not directly attached, then the routing table specifies the next router for that network. If the



**FIGURE**  
4.6

TCP delivers a byte stream by first decomposing it into fragments (called *segments*) that are transported by IP. TCP numbers the fragments, starting with some initial sequence number and then counting the bytes. IP divides the TCP fragments into packets. The IP header indicates the offset of the packet in the fragment.

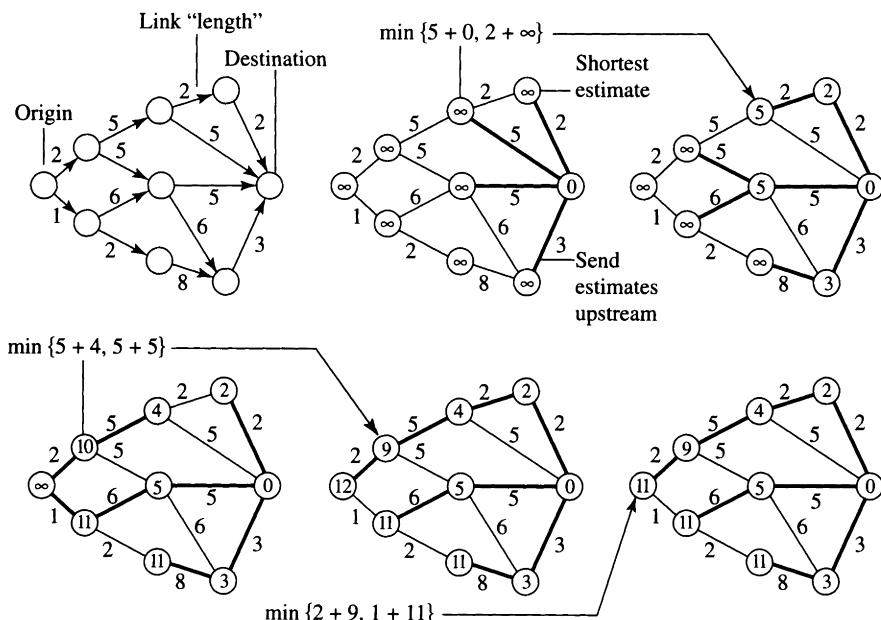
datagram is for a computer on the same network as the router, then it consults a table to find the physical (MAC) address of the computer.

To maintain the table with the addresses of computers attached on the same network, the router uses the *Address Resolution Protocol* (ARP). Using this protocol, a router searches its table for a given computer name whose MAC address it is trying to locate. If the computer name is not in the table, the router broadcasts a message on the network asking for the computer with the given address to reply. The computer with that address replies directly to the router, indicating its MAC address. The table entries have a time to live, and they are purged when that lifetime is exceeded. That is how the router updates the table entries to reflect changes in the network.

To maintain the routing table to the other networks, the routers typically use one of two versions of the shortest-path algorithm: the Bellman-Ford algorithm and Dijkstra's algorithm. We explain these two versions below. To implement that algorithm, the routers exchange control information such as their estimates of the length of a shortest path. In addition, the routers use a protocol called the *Internet Control Message Protocol* (ICMP) for exchanging

control information. (The term *gateway* is now replaced by *router*. A gateway is a node that passes data between devices with similar functions but dissimilar implementations. In this sense, a router is a layer 3 gateway.) ICMP provides a number of services that computers use for monitoring the network. One such service is *ping*, which asks a node to echo a packet, and *traceroute*, which obtains the route to the node and the delay on each hop over the route. Another service entails a router indicating to the source of an IP packet that the destination is unreachable. Similarly, a router indicates to the source when one of its packets exceeds its time to live. A router can also tell a source to redirect its packets in case of difficulties, or to slow down transmissions.

**Bellman-Ford Algorithm** Figure 4.7 illustrates a distributed algorithm that finds the shortest paths from all the nodes in a network to any particular node that we call the *destination* (RFC 1058). This is the *Bellman-Ford* algorithm.



4.7

FIGURE

Using the Bellman-Ford algorithm, the nodes find the shortest path to a destination. The nodes estimate the length of the shortest path to the destination. These estimates are shown inside the circles representing the nodes. When its estimate decreases, a node sends the new value to its neighbors. The figure shows, from left to right and top to bottom, the successive steps of the algorithm for the graph shown in the top-left diagram. Some intermediate steps are not shown.

The algorithm assumes that each node knows the *length* of the links attached to itself. These lengths can be a measure of the time taken for a packet to be transmitted along the links, or they can be arbitrarily selected positive numbers (e.g., 1 for every link). At each step of the algorithm, every node keeps track of its current estimate of the length of its shortest path to the destination. Whenever a node receives a message from another node, it updates its estimate. If the updated estimate is strictly smaller than the previous estimate, then the node sends a message containing the new estimate to its neighbors. A node  $i$  estimates the length of the shortest path to the destination by computing

$$L(i) = \min_j \{d(i, j) + L(j)\},$$

where the minimum is over all the neighbors  $j$  of node  $i$ . In this expression,  $d(i, j)$  is the length of the link from  $i$  to  $j$  and  $L(j)$  is the latest estimate received from  $j$  of the length of its shortest path to the destination.

Initially, all the estimates  $L(i)$  are set to infinity. At the first step, the destination reduces its estimate to 0. The destination then sends messages to its neighbors, who then update their estimates, and so on. Eventually, every node  $i$  finds out the shortest distance  $L^*(i)$  between itself and the destination. Moreover, there is a shortest path from  $i$  to the destination that goes first to  $j$  if and only if  $L^*(i) = d(i, j) + L^*(j)$ . Shortest paths need not be unique, and the nodes break ties arbitrarily. Figure 4.7 shows the convergence steps of the algorithm. The shortest paths are shown by thicker lines in the last step, at the bottom right of the figure.

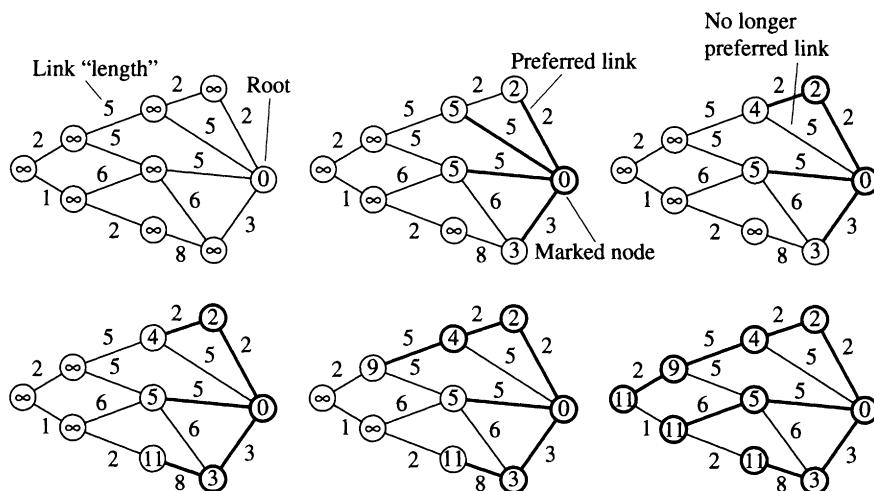
The shortest-path algorithm that we just explained is used by bridges for spanning tree routing. It is also used by the network layer of some networks to select good paths along which to route packets. Typically, these networks define the length of a link as a linear combination of the average transmission time and of the recent backlog in the queue of the transmitter of the link. By running the shortest-path algorithm periodically, the nodes can adapt their routing decisions to changing conditions. In this manner, the network reacts to link and node failures, and it also controls congestion. The algorithm is called a *distance vector* algorithm because each node sends to its neighbors the estimate of its distance to all nodes.

The Bellman-Ford algorithm is distributed: each node has only a partial knowledge of the topology of the network. It can be shown that the algorithm may fail to converge if the delay between updates is large and if during that time a significant amount of traffic is rerouted. To avoid these difficulties, the Internet started using a centralized shortest-path algorithm due to Dijkstra.

**Dijkstra's Algorithm** Consider once again the network shown in Figure 4.7. Assume that each node has a complete map of the network. To update the maps, each node sends to all the other nodes a message that indicates the lengths of the links attached to it. Algorithms that rely on global updates of local link information are called *link state* algorithms.

Each node then can implement a shortest-path algorithm to find the shortest path to every possible destination. The node could implement the Bellman-Ford algorithm that we discussed earlier. It can also implement the following algorithm due to Dijkstra.

Conceptually, the algorithm, called *open shortest path first*, or OSPF, by the Internet community, constructs a spanning tree of shortest paths from any given root node to the other nodes. Actual details of the algorithm depend on the implementation. The steps of the algorithm are shown from left to right and from top to bottom in Figure 4.8. To construct a spanning tree of shortest



4.8

FIGURE

A node uses Dijkstra's algorithm to construct a spanning tree of shortest paths from a node called the *root*. The network is shown in the top-left graph. We assume that each node has a complete map of the network. At each step, shown left to right and top to bottom, the algorithm picks the unmarked node with the smallest current label. The algorithm explores the neighbors of that node, after which it marks the node. The algorithm compares the label of each neighbor to the sum of the node's label plus the link's length. If that sum is smaller, it becomes the new label and the link is added to the list of preferred links while the previously preferred link leading to that node is removed from the list of preferred links. The steps leading to the last graph are not shown.

paths rooted at the “root” node, the algorithm first assigns a label 0 to the root node and  $\infty$  to every other node. The label of a node represents the length of the shortest path that the algorithm has discovered so far from the root to that node. At each step, the algorithm examines the node with the smallest label and explores its neighbors to see if it needs to reduce their label; in addition, the algorithm makes a node of the links that are along shortest paths. The algorithm starts with the root and examines its neighbors, that is, those nodes that can be reached in one step from the root. The algorithm compares the label of each of these neighbors with the sum of the root’s label (0) and the length of the link from the root to the node. If the sum is smaller, the algorithm replaces the label by that sum and notes that the link from the root to the node is on the shortest path discovered so far. The algorithm marks the root to remember that it has explored its neighbors. The algorithm then repeats the procedure with the unmarked node that has the smallest label. If the label of a node is reduced, then the new link that led to this reduction is noted as being a preferred link while the link that was previously on a shortest path is removed from the list of “preferred” links.

**Border Gateway Protocol** In IP, the routing uses a hierarchy. The network is decomposed into *autonomous systems* (ASs). An AS is a group of networks or subnetworks under a single administration. Routers used for information exchange within autonomous systems are called *interior routers*, and they use a variety of interior gateway protocols to accomplish this purpose. Routers that move information between autonomous systems are called *exterior routers*, and they use an exterior gateway protocol for this purpose.

Within each AS, the routing algorithm in each area router calculates the shortest paths to all the other routers in the areas to which it belongs using RIP, OSPF, or some other shortest path algorithm.

The routing between ASs uses the Border Gateway Protocol (BGP) (RFC 1771, 1772). ASs are attached together by one or more routers, called *border gateways*. Two ASs are connected if they share a link layer network that includes a border gateway from each AS. Each AS contains a router, called *border gateway speaker*, that implements BGP. BGP speakers use TCP (see section 4.4) to exchange routing tables and their updates. These routing tables specify the paths that each BGP speaker currently uses to reach other BGP speakers. BGP speakers exchange keep-alive messages to make sure that the paths are still valid.

Each BGP speaker compares these paths to construct preferable paths between itself and the other BGP speakers and sends these paths as updates to the other BGP speakers. This selection is left to each autonomous system. The selection must prevent the creation of loops and take into account restrictions

that ASs may have about carrying traffic from other ASs. Thus BGP operates on complete paths instead of summarizing them by their distance to the destination as the algorithms by Dijkstra or Bellman-Ford do. BGP enables autonomous systems to refuse to carry traffic originating from other given autonomous systems and also to favor specific paths for their own traffic. Each BGP speaker should remember the set of feasible paths, although it advertises only the preferable one.

### 4.3.2 Multicast IP

Multicast IP is a facility that sends an IP packet to a group of hosts identified by a group address (RFC 1112, 1584).

Multicast routing is implemented by a subset of IP routers called *multicast routers*. Group addresses may be either permanent or transient and are of class D (i.e., have 1110 as their higher order 4 bits). Each host can become or stop being a member of a group by sending join and leave request messages specifying the group address of the group it wants to join or leave. A multicast router learns which groups are active by periodically (every minute) polling the members of its domain. This polling process proceeds hierarchically using a protocol called the Internet Group Management Protocol (IGMP; RFC 1112).

Specifically, the multicast router sends a query, “which groups do you belong to?” to all the hosts of its local domain (using a group address to which all the multicast-capable hosts belong). On receiving the query, a host starts a count-down timer with a random value between 0 and some  $T$  ( $T = 10$  s is recommended). When the timer expires, the host sends a report with the group address as a destination address for each group it belongs to. The multicast router listens to all those reports destined to group addresses. If a host hears a report from another host—signaling that host’s membership to a common group—then the host does not send its own reply corresponding to that group. This procedure limits the number of reports per query. When it wants to join a new group, a host sends a report for that group, without waiting for a query. That report should be sent a few times at random intervals to make sure it gets to the multicast router. Note that the multicast router does not maintain a list of group members. It is required that hosts use their individual IP addresses, and not a group address, as the source address of the IP packets they send. For instance, the IP datagram reassembly algorithm assumes that each host uses a different source address.

The multicast routers then construct a tree of multicast routers from a host to the destinations that belong to the multicast group. IP tunnels (see next page) connect the multicast routers. This tree is the tree of shortest paths

from the source to all the possible destinations pruned back to cover only the destinations of the multicast group. To calculate that tree, the routers run the Bellman-Ford algorithm, explained earlier, where the destination is in fact the source of the multicast. In the pruned tree, the packets are duplicated at the last fork of the tree. Note that the pruned tree does not minimize the sum of the distances covered by the replicated packets. The results of the tree calculations are cached in the routers for subsequent packets.

As hosts join or leave a group, the routers automatically update the tree. If group members belong to the same LAN, the multicast router of the LAN converts the IP group address into a multicast LAN address. If group members belong to a common nonbroadcast multiaccess network (such as Frame Relay), then that network's multicast router must transmit copies to the different group members.

The *IP Multicast Backbone*, or MBone, is a network overlaid on the Internet, which is used for interactive video and audio multicasts. MBone comprises multicast routers, connected by virtual links formed by tunnels. Suppose a multicast packet arrives at MBone router A. A knows that it should be forwarded to MBone router B, which is not directly connected to A. The packet must be routed through a neighboring node C, which is not a multicast router. If A sends the multicast packet directly to C, C would not know how to forward it because the packet's destination address is not B's IP address but a multicast group address that C cannot understand. Therefore, A encapsulates the multicast packet in another IP datagram with destination address B and then forwards it to C. Eventually the packet reaches B where it is decapsulated and the original multicast packet retrieved. This encapsulation/decapsulation process creates a virtual link between A and B, called a *tunnel*, more precisely, a layer 3 tunnel since encapsulation is into a layer 3 or IP packet.

### 4.3.3 Reliable Multicast

Some multicast applications require a reliable delivery. Examples of such applications include the distribution of software, of newspapers, and of other documents. A number of mechanisms have been proposed for reliable multicast. We explain the main ideas behind these mechanisms.

Consider one source that multicasts a document to a large number of users. The first observation is that an acknowledgment-based scheme is not practical: If all the destinations were to acknowledge the packets they received, these acknowledgments would overwhelm the source. Also, merging acknowledgments at the branching nodes of the tree is not practical either because those nodes would need to count to make sure they got all the acknowledgments

before acknowledging to the upstream multicast node. Also, in such a merging scheme, if a user stopped listening, the source would keep on transmitting copies until the merging node learned that the user had left the multicast group.

If one packet gets dropped or corrupted on a link of the multicast tree, then a large number of users will not receive it. As a result, a negative acknowledgment scheme also risks overwhelming the source. However, it is possible to merge negative acknowledgments. Hence, all proposed schemes use negative acknowledgments with merging. Some schemes use “designated receivers” that act as caches. In such a scheme, every multicast node is assigned a receiver. When a negative acknowledgment reaches the node, it asks its designated receiver for a copy of the missing packet instead of propagating the negative acknowledgment upstream.

#### 4.3.4 Mobile IP

The objective of Mobile IP is to deliver packets to mobile nodes automatically. A mobile node is a host or a router that changes its point of attachment from one network or subnetwork to another. Using Mobile IP, a mobile node may change location without changing its IP address. (See [P96].)

A mobile node is associated with a fixed IP address, called its *home IP address*. A router on the mobile node's home network delivers IP datagrams to the mobile node when it is on a foreign network, that is, away from its home network. That router is called the *home agent*.

Two procedures can be used to deliver the datagrams to the mobile node on a foreign network. In the first procedure, the mobile node gets a temporary *care-of address* on the foreign network from some assignment mechanism and registers that address with its home agent. When the home agent gets a datagram for the mobile node, it encapsulates the datagram in an IP packet with the care-of address as destination address.

In the second procedure, the mobile node uses a *foreign agent*. The foreign agent is a router on the network visited by the mobile node. When it moves to another network, the mobile node registers with a foreign agent and obtains a care-of address from that agent. The foreign agent sends its address to the mobile node's home agent. The home agent encapsulates the datagrams destined to the mobile node in IP packets that it sends to the foreign agent. The foreign agent decapsulates the packets and delivers them to the mobile node with the care-of address.

Note that the first scheme does not require agents but is not automatic.

The encapsulation/decapsulation process is another example of tunneling. In either procedure, when returning home, the mobile node deregisters with

its home agent. Also, the mobile node uses its home IP address as the source address of its IP packets.

Agents (home or foreign) advertise their availability by periodically sending *agent advertisement messages*. Moreover, mobile nodes may solicit such a message by sending an *agent solicitation message*. As it receives these messages, the mobile node can determine whether it is on its home network or on a foreign network. On its home network, the mobile node operates without the mobility services of Mobile IP, that is, by using the standard IP.

The advantage of the first procedure is that it does not require foreign agents. Its disadvantage is that networks must maintain a pool of addresses available for visiting mobile nodes.

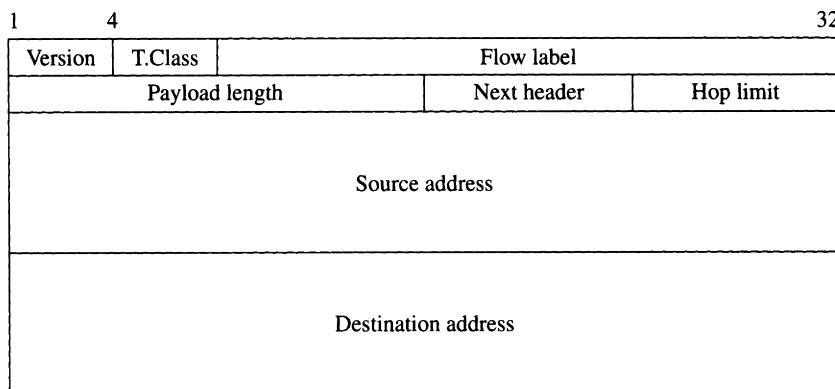
The mobility management messages are authenticated to prevent redirection attacks, such as a third party stealing messages by giving its address as the new mobile node address.

### 4.3.5

### IPv6

The current version of IP is version 4. Its main limitations are a limited number of addresses, a complex header, a difficult mechanism for introducing extensions and options, a limited number of different services, and poor security and privacy. IPv6 is being designed to overcome these limitations. The transition from IPv4 to IPv6 is expected to take many years. In the meantime, the two versions can coexist by tunneling: IPv4 forwards IPv6 packets without modification. (For details, see RFC 2460, "IPv6 specification.") An IPv6 version of ICMP is designed (RFC 2463). The IPv6 header consists of a 40-byte header followed by up to six optional extension headers.

The 40-byte header is shown in Figure 4.9. The version field contains the binary representation of the number 6, indicating that the packet is an IPv6 packet. The router uses the traffic class field to determine the importance and urgency of the packet, somewhat like the TOS field can be used in IPv4. The flow label may be used to describe the characteristics of the traffic to which the packet belongs. The payload length indicates the number of bytes in the packet that follows the header; zero here indicates that the actual packet length is specified in a hop-by-hop extension header (see next page). The next header indicates which of the six extension headers follows this header, if any, and whether the packet is for UDP or for TCP, otherwise. The extension header has a similar next header field. The hop limit is decremented by one by each router to prevent packets from looping forever in the network; the packet is discarded when the value reaches zero, so that the maximum number of hops is 254. The source address and destination address have 16 bytes (RFC 1884), enough for



4.9

**FIGURE**

The IPv6 header consists of the 40-byte header shown here, followed by up to six extension headers.

$5 \times 10^{28}$  addresses per human being, which should suffice for a while; there is no explicit support for mobile hosts. Note that there is no error-detection field in the header: IPv6 relies on higher-layer protocols for error control. In IPv6, the routers do not fragment packets. When a packet is too long for a router to handle, the router sends an ICMP message to the source asking it to fragment the packet and resend it.

The extension headers are the following:

- ◆ *Hop-by-hop Options*: Information that every router must examine (usage to be determined later).
- ◆ *Routing*: Specifies a set of routers that the datagram must visit (a list of up to 24 IPv6 addresses).
- ◆ *Fragmentation*: This header specifies which fragment of a larger packet this IPv6 packet contains. By using this extension header, the destination can reassemble the packet as in IPv4.
- ◆ *Authentication*: This header contains a checksum that enables the destination to authenticate the sender.
- ◆ *Encapsulating Security Payload*: This header contains the encryption key number and the encrypted payload. The destination can recover the original payload from these two items using a decryption algorithm that is left to the choice of the user.
- ◆ *Destination Options*: The possible use of this header, intended only for the destination, has not been defined so far.

This header structure is very flexible in that it leaves the door open for extensions. It is also rather efficient in that the minimal header is simpler than in IPv4 by the removal of the checksum and of fragmentation. With time, the Internet designers will develop mechanisms that exploit this format to implement a large range of services.

#### 4.4

#### TCP AND UDP

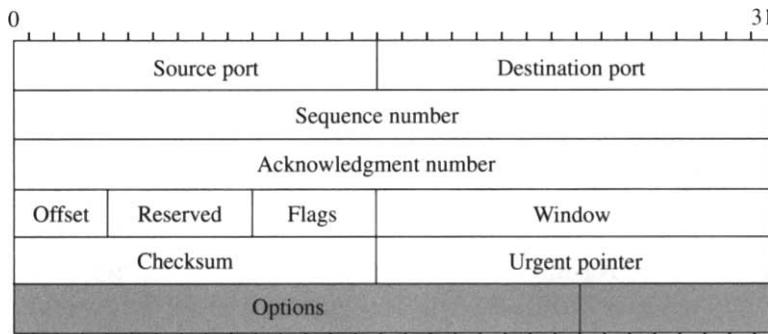
The Internet uses two different transport layers: UDP and TCP. UDP and TCP data units from the transport layer are forwarded to the IP layer, which encapsulates them in IP packets.

UDP, the *User Data Protocol*, is a connectionless service that provides for multiplexing and for error detection, without guaranteed delivery or duplicate prevention. UDP is useful for simple query-response applications because no time is lost for TCP connection establishment. UDP is also used for applications where reliability is not critical or where retransmission delays are unacceptable. For instance, telephony over IP uses UDP, as do audio and video streaming applications. The UDP header consists of 16-bit source and destination port numbers, a 16-bit length field that gives the total length (data and header) in bytes, and a 16-bit checksum over the header and data (RFC 768).

TCP, the *Transmission Control Protocol*, is a connection-oriented, duplex, reliable, byte-stream service with flow control. *Byte-stream service* means that within a TCP connection, the source sends a sequence of bytes for delivery to the destination. TCP buffers a group of these bytes into a *segment* and gives it to IP. The size of the buffer is *MSS*, or maximum segment size. The default value of MSS is 536. A segment has no semantic significance. *Reliable service* means guaranteed delivery, in order, and without duplicates. TCP uses the Go Back N protocol (GBN), discussed in section 2.6.3, to ensure reliable service. A version of TCP, SACK TCP, uses the Selective Repeat Protocol (SRP).

Figure 4.10 shows the TCP header (RFC 761). The 16-bit source and destination port numbers permit multiplexing of several connections between hosts. Well-known ports, such as 21 for FTP and 80 for HTTP, are fixed.

Every byte of data sent over a TCP connection has a 32-bit sequence number given by the sum of the segment sequence number and its position in the segment. The sequence number of the first segment of a connection is agreed on by a three-way handshake: (1) the source announces the sequence number; (2) the destination acknowledges that number; and (3) the source sends the first segment with that number. This procedure prevents the misunder-



4.10

**FIGURE**

TCP header. This header indicates the source and destination port numbers. The header also contains the sequence and acknowledgment numbers needed by the SRP that TCP implements. The window size specified by the header is negotiated by the hosts.

standing that would be caused by a delayed packet. Connections are released by one two-way handshake for each direction.

The 32-bit acknowledgment number is cumulative so that an acknowledgment of sequence number  $n$  indicates that all bytes up to but not including  $n$  have been received, and  $n$  is the next expected byte number. (Thus, if sequence number  $n$  is acknowledged three times, say, the sender may infer that the packet containing that byte is lost.)

The 4-bit offset gives the number of 32-bit words in the TCP header, including options, and indicates where the data begins.

Six flags may be set: URG, ACK, PSH, RST, SYN, FIN. If URG is set, the 16-bit urgent pointer indicates the position of the first byte of nonurgent data in the segment. (Presumably, all the preceding bytes are urgent.) ACK is set once the connection is established, indicating validity of the acknowledgment number. PSH is set to indicate that the send buffer was emptied by the packet containing  $PSH = 1$ . When set, RST immediately terminates the connection. SYN is set to establish a connection, indicating that the segment carries the initial sequence number. FIN is set to request normal termination of the connection: the two-way handshake requires a FIN in each direction. The checksum over the data and header provides error detection.

Thus, the steps of a TCP connection from  $A$  to  $B$  are as follows:

1.  $A$  sends a SYN to  $B$  to indicate that it wants to open a connection. The connection is identified by the source and destination IP addresses and TCP port numbers, as well as by an initial sequence number that  $A$  determines

from its clock. This sequence number prevents confusion that might be caused by delayed SYN packets.

2.  $B$  sends a SYN.ack back to  $A$  to acknowledge the start of the connection and to indicate the initial sequence number it will use when sending to  $A$ .
3.  $A$  sends the first data packet, which also acknowledges the reception of the SYN.ack. This packet completes the three-way handshake.
4.  $A$  keeps sending packets and  $B$  acknowledges every correct packet it receives with an ACK whose sequence number is that of the next byte that it expects to receive.  $A$  and  $B$  use the GBN protocol for this exchange. (Note that in many TCP implementations, the receiver delays the transmission of acknowledgments.)  $A$  adjusts the window size as we explain in section 4.6.
5. When one of the hosts, say  $B$ , wants to close the connection, it sends a FIN to the other host  $A$ .  $A$  then sends back a FIN.ack to complete a two-way handshake close of the connection from  $B$  to  $A$ . Finally,  $A$  sends a FIN to  $B$  and  $B$  replies with a FIN.ack to close the other side of the connection.

The options field is used at connection establishment to negotiate a variety of options, including MSS.

The 16-bit window,  $W_{max}$ , is used for flow control. It is set by the receiver at the maximum number of unacknowledged bytes it is willing to accept from the sender. It is usually set at the size of the receiver's buffer that holds out-of-order packets. If the buffer is full, the receiver may reset the window to zero. The last byte the sender can send is the acknowledgment sequence number plus the window size. At any time, the source must use SRP with window size equal to the smaller of  $W_{max}$  and the window size  $W$  calculated by a congestion avoidance algorithm, studied in section 8.3.4. A number of such algorithms have been proposed, and we discuss their main operations in section 4.6.1. Roughly,  $W$  is increased when congestion is not indicated and decreased when the source detects congestion.

#### 4.4.1 Applications

On top of the TCP layer are generic applications, as shown in Figure 4.2. We describe some of these.

An application uses TCP or UDP via system calls that are part of the operating system. For instance, in UNIX and in Windows, the system calls are socket-based. A *socket* can be thought of as a queue attached to the TCP or UDP protocols. To give a sense of how these system calls are used, imagine that you want to send messages from a computer with IP address IP1 to a computer with

IP address IP2. In one case, we send a message stored in buffer B1 using UDP with source port number P1 and destination port number P2. The computer IP2 writes the message into a buffer B2.

The pseudo-code at IP2 is as follows:

```
create socket S2 % create the socket object
bind S2 to IP2.P2 as UDP % attach the socket to UDP at IP2 and P2
receive S2 into B2 % write what arrives at the socket into B2
close S2 % delete the socket after the transaction is completed
```

Here is the pseudo-code at IP1:

```
create socket S1 % create the socket object
bind S1 to IP1.P1 as UDP % attach the socket to UDP at IP1 and P1
send B1 to IP2.P2 through S1 % write the message from B1 to the socket
close S1 % delete the socket
```

Now consider the case of a TCP connection from IP1 with port number P1 to IP2 with port number P2.

The pseudo-code at IP2 is as follows:

```
create socket S2 % create the socket object
bind S2 to IP2.P2 as TCP % attach the socket to TCP at IP2 and P2
listen to S2 % get ready to receive packets
accept % accept a connection
read S2 into B2 % write into B2 what you receive on S2
close S2 % terminate the connection
```

Here is the pseudo-code at IP1:

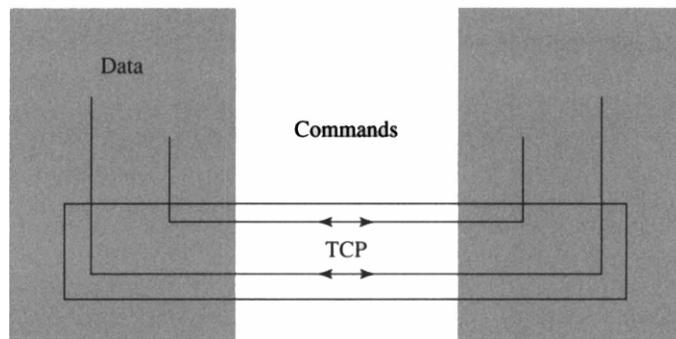
```
create socket S1 % create the socket object
bind S1 to IP1.P1 as TCP % attach the socket to TCP at IP1 and P1
connect S1 to IP2.P2 % open the connection
send B1 to S1 % write the messages from B1 to the socket
close S1 % delete the socket
```

The actual code depends on the operating system. Many textbooks provide examples.

#### 4.4.2 FTP

FTP, the *File Transfer Protocol*, enables users to transfer files between computers. As Figure 4.11 shows, FTP opens two connections between the computers:

Interactive: send, get, transfer, cd  
Modes: stream, block, compressed



4.11  
FIGURE

The File Transfer Protocol (FTP) sets up two connections: one for the commands, the other for the data exchange.

one connection for the commands and replies and the other for the data transfers. FTP is interactive. Its commands are *send*, *get*, *transfer*, and *cd* (change directory). FTP transfers files in three modes: *stream*, *block*, and *compressed*. In the stream mode, FTP handles information as a string of bytes without separating boundaries. In the block mode, FTP decomposes the information into blocks of data. In the compress mode, FTP uses the Lempel-Ziv algorithm to compress data.

#### 4.4.3        SMTP, rlogin, TFTP,                 and HTTP

The first two of these applications run on top of FTP, the third is based on UDP, and the last on TCP.

The *Simple Mail Transfer Protocol* (SMTP) is used for e-mail. SMTP accepts messages with a list of destinations. When it does not succeed in delivering a message, SMTP retries a number of times, and it notifies the sender if it cannot deliver the message.

The *remote login service*, rlogin, enables a user to access a remote machine. Rlogin establishes the connection to the remote machine, exchanges the required authorization information, and eventually logs in the user. When rlogin is used, the echoing is performed by the remote machine. That is, the characters

typed by the user are sent back by the remote machine before being displayed. Special control commands such as *stop* and *resume* are sent as urgent data.

The *Trivial File Transfer Protocol* (TFTP) transfers data as blocks of 512 bytes. TFTP sends one block of 512 bytes and waits for an acknowledgment. TFTP retries after a timeout until it succeeds and then proceeds to the next block. TFTP numbers the blocks sequentially from 1. This robust protocol operates even when the transport layer is of low quality. However, it is not efficient and is useful only when a lightweight protocol is needed, for instance as a simple boot ROM application for loading kernels over the network.

The *Hypertext Transfer Protocol* (HTTP) is the basis for access over the World Wide Web to resources referenced by their URL. The success of the Web is virtually due to the flexibility of HTTP. The HTTP protocol is a request/response protocol on top of TCP. A client, often a browser, opens a TCP connection with a server and sends a request in the form of a method (e.g., GET, POST, HEAD, DELETE), the URL of the resource (the object to which the method is to be applied), and the protocol version (e.g., HTTP/1.1), possibly followed by modifiers. The server responds with a status line, including the message's protocol version and a success or error code (e.g., OK, Payment required) followed by a message containing server information, and possibly, a body giving information about the data (e.g., how it is coded) and the data itself (e.g., an HTML document). The browser must be able to interpret the data. In the first version of HTTP, the response was a message in hypertext mark-up language (HTML). In the current version, the form of the HTTP request and response is extensible so that new methods and resource types (images, video) can be introduced. The complexity of browser programs is due to the large number of methods and resource types and display formats that must be handled.

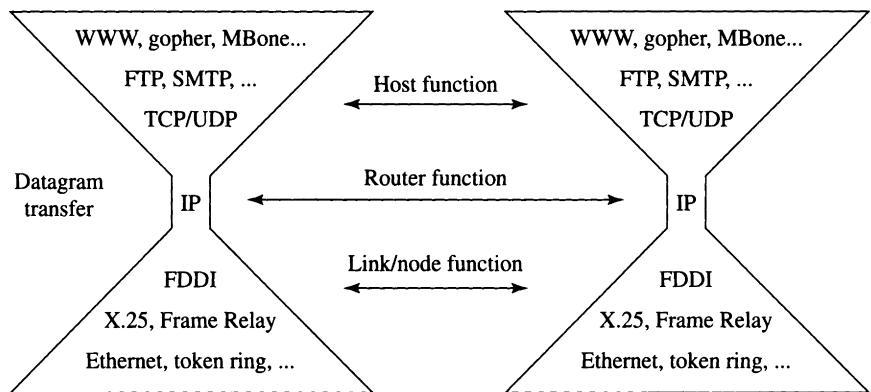
---

## 4.5

## INTERNET SUCCESS AND LIMITATION

We studied the Internet suite of protocols in terms of the variety of services that it supports. We also saw that those protocols are organized in a multilayered architecture resembling the seven-layer OSI model. Figure 4.2 summarizes both of these aspects. It lists the different protocols in a manner that shows their layered dependence and how those layers are related to the OSI model.

In order to explain the technical basis for the astounding success of the Internet and to foresee its limitations, it will be more revealing to view the



**4.12**  
**FIGURE**

The Internet implementation of the Open Data Network model shows a simple bearer service, supported by many networks and supporting a variety of applications. The bearer service is provided by simple routers, facilitating network interconnection. Applications are provided by hosts, facilitating experimentation and incremental evolution.

Internet protocols as an implementation of the Open Data Network model of section 2.8. Such a view is presented in Figure 4.12.

The narrow “waist” of the figure represents the single and simple IP bearer service: the end-to-end transport of IP datagrams. This service can be provided by routers capable of the basic tasks of storing packets and routing them to appropriate outgoing links after consulting a routing table. Most significantly, there are no performance requirements, so that slow and old routers can coexist with new, high-performance routers. This “backward compatibility” is enormously valuable and explains why networks with widely differing performance can be interconnected (see Figure 1.13). Domain names and IP addresses are assigned on a decentralized basis, so network growth is opportunistic rather than planned. The benefits of connecting to the Internet grow with its size at the same time as equipment costs (LANs, links, computers) decline because of scale economies, resulting in a doubling of the size of the Internet each year.

The brilliance of the IP design lies in its simplicity: because IP datagrams are self-contained, routers do not need or keep any state information about those datagrams. As a result, the network becomes very robust. If a router fails, datagrams in the router may be lost, but new datagrams would automatically be routed properly, with no special procedures. (By contrast, if the bearer service were connection-oriented like ATM, routers would have to maintain connection-state information, and if a router were to fail, that state information

would be lost, making it very difficult to restore the connection.) Simple rules in routers can help route traffic around congested parts of the network, giving the network the capability to adapt to changes in traffic.

The third feature of the figure—its wide top—represents the rich variety of applications that the IP bearer service, together with UDP and TCP, can support. The last decade of successful, sophisticated applications like the World Wide Web has shown that UDP/IP datagram service and TCP/IP reliable, byte-stream can serve as building blocks for complex services, provided that the end hosts are sophisticated.

The most remarkable aspect of this feature is that the application software resides entirely in the end hosts and not in the routers. This means that the same basic service, implemented by simple routers, can support these sophisticated applications. Thus the network hardware and software have a much longer technical and economic life than does the end host. Indeed, in the mid-1990s, significant numbers of Internet routers are 15 to 20 years old. This also implies that parts of the Internet can experiment with new applications on advanced hosts using the real network, while other parts of the Internet continue undisturbed to run old applications on primitive hosts. This ability to experiment with new applications has greatly helped the proliferation of new applications.

Thus the technical basis for the Internet's success is its reliance on simple routers to transfer individual datagrams and on advanced end hosts to run sophisticated applications. The simple infrastructure is compatible with a wide range of applications. The developers of successful applications can distribute them for fame or profit, without requiring any change from the infrastructure. The contrast with the telephone network could not be more striking. There the network “intelligence” is located in its expensive switches, while the end hosts (the telephone sets) are primitive, with little functionality. The introduction of new services requires changes in the infrastructure—changes that are slow and expensive. Hence experiments are costly and infrequent, and new services are introduced after much deliberation and planning.

The limitations of the Internet can be foreseen from the figure. The IP bearer service cannot provide any guarantees in terms of delay or bandwidth or loss. Routers treat all packets in the same way. (This “equal service for all” is, perhaps ironically, called *best-effort service*.) This is an innate feature: the absence of state information means that packets cannot be differentiated by their application or connection, and so routers would be unable to provide additional resources to more demanding applications.

The technical challenge is to expand the services offered by UDP/TCP and IP to provide guarantees in a way that preserves the Internet's accommodation

of backward compatibility and incremental change. We now discuss some recent proposals to meet this challenge. The proposals can be roughly classified in two types: those that assume IP bearer service but expand the services offered by UDP and TCP to better suit applications; and those that alter the stateless nature of IP routers. Most proposals are of the first type.

UDP service offers unreliable delivery of individual packets with little overhead, and TCP service offers reliable, ordered flow of byte streams at the cost of considerable overhead. Application requirements may not readily map into either of these two services. For example, video streaming applications may be better matched to a service that offers unreliable, ordered flow of byte streams. Some HTTP applications, on the other hand, may be better matched to a reliable, unordered, delivery of messages.

## **4.6 PERFORMANCE OF TCP/IP NETWORKS**

We study the performance of TCP/IP networks in section 8.3.4. In this section, we highlight the main performance characteristics of these networks because they justify a number of modifications that are being proposed for the TCP/IP protocols.

We first consider the window adjustment mechanism of TCP. This mechanism determines the traffic that TCP sources produce. In 1999, about 85% of the traffic on the Internet is TCP. Our discussion shows that TCP is biased in favor of connections with a small round-trip time. To correct this bias, variations of TCP have been proposed and so have modifications of the routers. Next, we examine the notion of differentiated service and discuss possible implementations either in TCP or in IP.

### **4.6.1 Window Adjustment in TCP**

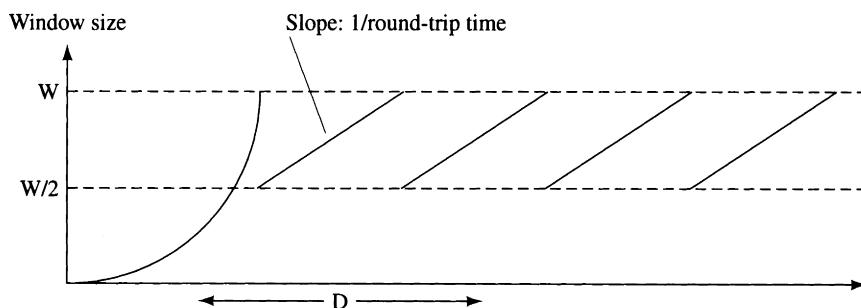
The transmission rate of a TCP source depends on the window size. Recall that the window size is the maximum number of bytes that the source can send in a set of packets for which it has not received acknowledgments. As we explained, the destination computer sends back an acknowledgment for every correct packet that it receives. Assume that the round-trip time of the connection is  $T$  seconds. That is,  $T$  is the time from the transmission of a packet until the reception of its acknowledgment by the source. If the window size is

$W$  bytes, then the source sends  $W$  bytes in  $T$  seconds, and its transmission rate is  $W/T$  bytes/s.

It is tempting for the source to choose a very large value for  $W$  to increase its throughput. However, doing so makes it likely that a router will not be able to transmit as fast as the source. Ideally, the source should adjust its window size so that it gets its fair share of the transmission rate of the routers. The precise meaning of "fair share" can be debated, but a simple interpretation is that if  $N$  connections share a router that is their only bottleneck and transmits with rate  $C$ , then each connection should be able to transmit with a rate close to  $C/N$ . It would be very unfair if some connections could transmit at a much larger rate than the others. Also, if the window sizes are too small, then the connections are not taking advantage of the link capacities. Summing up, we see that the objectives of the window adjustment mechanism are to share the links fairly and efficiently among the connections.

The TCP window adjustment mechanism starts by increasing the window size exponentially fast to "discover" the available transmission rate. When the source fails to get an acknowledgment, it suspects that a router has dropped a packet and it reduces its window size by 50%. From that point on, the source increases its window size by one unit every round-trip time. The algorithm then continues as before. Figure 4.13 shows the resulting evolution of the window size of a connection that goes through a router, assuming that it is the only active connection.

From this simple graph, we can infer the following conclusions. First, a connection with a small round-trip time is much more aggressive than one with a large one. Consequently, if connections with different round-trip times share a router, the ones with the smaller round-trip times are likely to grab most of the link capacity. We elaborate on this observation in section 8.3.4.



4.13

Window size of a single TCP connection going through a single router.

FIGURE

Second, note that TCP loses approximately one packet every  $D = TW/2$  seconds where  $T$  is the round-trip time (see figure). In  $D$  seconds, the connection sends approximately  $W/2 + (W/2 + 1) + \dots + 2W/2 \approx 3W^2/8$  units. Assume that one unit is  $K$  average packets. Thus, if we designate by  $R = 3KW^2/8D = 3KW/4T$  the throughput of the connection (in packets/s) and by  $L = 1/(3W^2/8) = 8/3W^2$  its loss rate, then we find that

$$R = \frac{1.25K}{T\sqrt{L}}.$$

That identity shows that the throughput is inversely proportional to the square root of the loss rate. This observation is sometimes used to determine whether a proposed protocol is "TCP-friendly." If the throughput of the protocol is larger than that of TCP for the same loss rate, then one decides that the protocol is not TCP-friendly, which is to say that it is grabbing an unfairly large fraction of the router bandwidth. Of course such a summary judgment neglects the effect of interactions between protocols. More importantly, it decides that a protocol that is "better" than TCP is not desirable.

Observe also that if the link from the destination to the source is slow, then it may limit the rate at which the destination can acknowledge packets, which in turn limits the rate at which the source can send packets.

When a noisy link (e.g., wireless) drops packets because of transmission errors, TCP slows down because it assumes that the losses are due to congestion. As a result, the performance of TCP is very poor when links are noisy.

## 4.6.2 Suggested Improvements for TCP

To improve the behavior shown in Figure 4.13, Internet researchers suggested the following improvements for TCP:

1. Increase the window size at a rate that is independent of the round-trip time.
2. Base the window adjustment mechanism on delays, not on losses.
3. Have the routers mark the packets to indicate that they are getting congested instead of waiting until they have to drop packets.
4. Base the decision to drop packets on the number of packets of the same connection in the router, not simply on the total number of packets.
5. Have the routers provide the sources with a more explicit indication of the available transmission rate.

6. Merge acknowledgments on a slow reverse link.
7. Introduce a link error control mechanism for noisy links.
8. Cheat and modify your TCP window adjustment mechanism.

The first modification would eliminate the bias in favor of connections with shorter round-trip times. However, experiments show that it is not easy to choose a rate of increase of the window size that works well over a wide range of round-trip times. If that rate of increase is small compared with the current rate, then connections are slow. If the rate is excessive, then the protocol loses too many packets when used by connections with a large round-trip time.

The second proposal consists in comparing the round-trip time with the minimum round-trip time observed over many successive packets. The idea is that the increase in round-trip time measures the amount of queuing in the routers. TCP-Vegas is a protocol that uses this mechanism. The practical difficulty is that the estimated minimum round-trip time includes the queuing time at the start of the connection. If all the sources were to use this mechanism, they might keep on increasing the congestion.

The third mechanism (called *explicit congestion notification*) requires that the routers be able to mark packets and that the receiver send back the marks in the acknowledgments. This mechanism is desirable because it signals a source to slow down without paying the penalty of forcing retransmissions. The actual adjustment of the window size based on the marks must be done carefully to avoid the bias in favor of connections with a short round-trip time.

The fourth proposal could limit the bias of TCP. Unfortunately, this mechanism is more complex to implement. It requires that the router be able to keep track of connections and to count packets of the different connections.

The fifth proposal necessitates a sophisticated router that can provide useful indications to the sources. One proposal is to mark a packet if it is likely to cause a future loss of a packet. That is, as a packet arrives, the router predicts whether this packet will force it to drop one more packet that will arrive later. Such decisions must be based on the current load of the router.

The merging of acknowledgments on a slow reverse link, for instance by having one ACK sent for every  $N$  packets, can eliminate the effect of the asymmetry of the rates.

The link error control protocol converts an unreliable link into a reliable one, which improves the performance of TCP.

Unfortunately the last proposal is implemented in a number of commercial products that proudly advertise better network performance. It is easy to cheat

in the TCP implementation by increasing the initial window size, the slope of the window increase rate, or the factor by which the window size is decreased.

### 4.6.3 Suggested Improvements for IP

One major advantage of IP is its simplicity, which makes the network scalable and the equipment easily upgradable. The vision is that if one builds a very fast network, performance problems go away. For instance, if all the links work at 10 Gbps and have a load of less than 90%, then the average delay per node should be less than 10 packet transmission times, or less than 80  $\mu$ s for 1-KB packets. If a path across the Internet goes through at most 20 nodes, the average end-to-end delay should be less than 1.6 ms plus the unavoidable propagation delay. This level of performance is more than adequate for conversation and interactive applications.

There are a few difficulties with this vision. The first one is that many links will not be operating at 10 Gbps for a number of years. That is the case in particular for the access links that connect the user's LAN or computer to the ISP router. The rate of such links is typically below 1 Mbps. Another problem is that the sources of TCP traffic are greedy and try to grab all the available bandwidth. These sources create bottlenecks by sending bursts of packets until losses tell them to slow down. A third problem is that the demand for bandwidth grows with the supply. As the network performance improves, the users derive more benefits from using the network and they tend to use it more. Also, applications that make use of the improved performance appear quickly and increase the network load.

In view of these difficulties, a number of methods have been designed to guarantee the level of performance that some applications require. These methods involve separating different classes of traffic and handling them differently in the routers.

### 4.6.4 Queuing Algorithms

A packet scheduler sorts packets that must be forwarded through a link interface into separate queues according to their QoS class. The algorithms attempt to provide the requested QoS by appropriately scheduling the transmission of packets.

In *first-come first-served* or FCFS queuing, packets are transmitted in the order they are received. This widely used algorithm makes no distinction

among packets with different QoS requirements. Sources that deliver more or larger packets will receive greater bandwidth and may inflict large delays on sources that deliver fewer packets.

In *priority queuing*, packets are sorted into different priority queues. Higher priority packets are transmitted ahead of lower priority packets. When there is a large volume of higher priority traffic, lower priority traffic may get very little bandwidth and suffer large delays.

In *class-based queuing*, packets are sorted into queues, one queue per class. Packets within each queue are transmitted FCFS. Different queues are served in round-robin fashion. The number of packets transmitted from each class, or the bandwidth devoted to each class in each round, is determined by the attributes of that class. Thus the algorithm guarantees that each class will receive a certain fraction of the link bandwidth. Such guarantee is not provided by the two previous algorithms. However, within a class, “misbehaving” sources that deliver more or larger packets will inflict delays on sources that deliver fewer or smaller packets. In this sense, the scheduling algorithm may be unfair.

*Fair queuing* is a term applied to a set of algorithms that attempt to allocate bandwidth fairly among all flows within a certain class. A flow may be defined, for example, as data flowing between a source-destination pair, and the class as a collection of such flows. A fair bandwidth allocation might then be defined as one in which low-volume flows within the class get their entire requested allocation, whereas high-volume flows share the remaining bandwidth equally.

Some queuing algorithms are analyzed in Chapter 8. The QoS experienced by a flow depends on the scheduling algorithm in all the switches in its route. Since the IP layer forwards individual datagrams, packets within a flow cannot be assigned a fixed route and so it is difficult to guarantee QoS within an IP network. One advantage of circuit- and virtual circuit-switched networks is that they assign routes to flows and can better provide QoS guarantees. IP switching may also provide those guarantees.

Queuing algorithms that allocate bandwidth among different queues can be augmented by *active queue management*, in which packets may be selectively discarded. Such a mechanism may delete a lower priority packet from a queue when a higher priority arrives and the queue is full.

#### 4.6.5 Label Switching

With increasing link speeds, limits on router throughput begin to constrain network growth. Routers are slower than layer 2 (Ethernet, ATM) switches, because the look-up of large routing tables is time-consuming. One recent advance seeks to bring to routers the speed of layer 2 switches for routine

packet forwarding, while maintaining the capability that routers provide in terms of protocols, packet processing, and security. This involves a technique called *label switching*.

The key idea to label switching is that the incoming packet carries a label that is directly translated by the switch hardware into the number of the switch's outgoing port or interface. Implementing this idea in routers involves two components. First, packets within a particular flow must be bound to a label that is quickly translated (in hardware or software) into the router's outgoing layer 2 interface. Second, an algorithm must distribute labels among the routers so that a consistent route is assigned to the flow. (It would be bad if, for instance, the route created loops.) By binding a flow to a label, and a label to a route, all packets in the flow follow the same route. Thus, this technique achieves some of the benefits of virtual-circuit switching. In particular, it provides the opportunity to guarantee QoS to a flow.

We give a highly abbreviated description of two implementations. One is called *IP switching* (RFC 1953); the other is *tag switching* (RFC 2105) with a variation called *multiprotocol label switching* (MPLS, RFC 2547). Either technology can be deployed within a subnetwork of appropriately equipped routers.

IP switching requires an underlying ATM link layer, so the label attached to a particular flow corresponds to an ATM virtual circuit/path identifier (VCI/VPI) (see Chapter 6). The flow classification is based on a TCP/UDP port pair (all packets with the same port pair comprise the same flow) or, with a greater level of aggregation, on a source-destination host pair. A label distribution protocol is invoked each time a new flow is identified. Because there is significant overhead attached to the protocol, it is counterproductive to identify short-lived flows. Routers employ a heuristic to determine if a potential flow will be long-lived. The router counts the number of packets belonging to a potential flow that arrive within a short time. If that number exceeds a threshold, the flow is identified and the router invokes the label distribution protocol. The protocol distributes the flow-label binding from router to neighboring router. If the number is below the threshold, the router forwards the IP packets in the normal manner. The binding between a flow and its label is valid for a short time (60 seconds, say), so that the VCI/VPI can be assigned to another flow.

In tag switching, a short fixed-length tag is attached to packets in a flow in a way that depends on the underlying link layer packet format. The link layer need not be ATM. Routers at the entrance to the tag-switched subnetwork insert the tag, and routers at the exit remove the tag. Flow classification is more flexible than in IP switching. (For example, a multicast flow can be assigned a tag, based on its class D address.) Like the VCI/VPI labels used in IP switching, tags are local to links, so when a packet leaves a router, it may

acquire a new tag. The tag to output interface binding is placed in a table. Table access is faster than with routing tables because there are fewer, fixed-length entries, which contributes to the greater throughput of tag-switched routers. Unlike in IP-switching, flow classification and tag distribution are carried out by router protocols similar to other interior gateway protocols. Also, unlike in IP-switching, the tag assignments may be permanent.

#### 4.6.6 Suggested Improvements for Other Protocols

##### *RSVP*

RSVP (Resource Reservation Protocol) is a resource reservation setup protocol designed for multicast, multimedia data streams or flows (RFC 2205). A flow is specified by attributes such as source-destination pair, mean data rate, latency, and quality of service (QoS) (RFC 1363). For example, an MPEG video stream that averages 3 to 7 Mbps and peaks to 12 Mbps at 30 frames per second, requires certain QoS for adequate reception. An application signals to the network its QoS requirement, and the protocol reserves resources in the network switches.

A source sends a message (called *path message*) that gives the multicast destination address, the flow specification, and a template for identifying which packets belong to the flow.

Receivers join a multicast tree by sending messages (called *reservation messages*) with a description of the QoS they expect. The routers forward upstream a least upper bound of the QoS requirements made downstream. Thus, if a router gets a request from a downstream node for a 200-ms delay version of a stream and another from a different node for a 100-ms delay version, the router forwards the 100-ms delay request upstream. (The precise meaning is admittedly fuzzy.) Once the router gets the 100-ms delay version from upstream, it sends it to the downstream nodes. A similar idea is used for throughput requests. If a router does not find enough spare resources to carry the stream with the requested QoS, it sends a reject message to the receiver. Presumably, a switch algorithm uses these requests for QoS of different streams to determine the resources that should be reserved and how to schedule the transmissions of packets.

To adapt to changes, RSVP specifies that sources periodically send messages that describe their streams and receivers periodically send reservation requests. The downstream messages sent by the source are routed by multi-cast IP. The reservation messages are sent upstream along the reversed tree.

All these messages carry a timeout value that the nodes use to set timers and to delete the corresponding information when the timers expire.

RSVP mechanisms needed to implement QoS requests include (1) a *packet classifier* that determines the QoS class of each packet in accordance with the reservations that are made, (2) *admission control* that determines whether the node has sufficient resources to provide a requested QoS, and (3) a *packet scheduler link-layer mechanism* that achieves the requested QoS using one of several queuing algorithms.

*Real-time protocol* or RTP provides unreliable, end-to-end delivery services for unicast or multicast data with real-time characteristics, such as interactive audio and video (RFC 1889). Those services include payload type identification, sequence numbering, time stamping, and delivery monitoring. Applications may run RTP on top of UDP to make use of its multiplexing and error-indication services. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence, but sequence numbers might also be used to determine the proper location of a packet, for example in video decoding, without necessarily decoding packets in sequence. Applications that run on top of RTP include VIC, a real-time, multimedia application for video conferencing over the Internet, and VAT, an audio conferencing application.

The widespread use of the HTTP/1.0 protocol over TCP has created problems. Typical Web pages today contain an HTML document and many small embedded images. Since HTTP/1.0 opens and closes a new TCP connection for each operation, this practice means a high fraction of packets are simply TCP control packets used to open and close a connection. The creation of many parallel, short-lived TCP connections, and their interaction with TCP congestion control algorithms, leads to poor performance and congestion. We summarize several proposals to ameliorate these problems.

Transactional TCP or T/TCP (RFC 1644) is designed for client/server interaction in which a request is followed by a single response. The client and server cache a connection count number (CC). A new SYN TCP client request packet includes the new connection count and data (request). The new CC is larger than the cached CC, and so the server knows that this is a new request and not an old, delayed packet. The server's reply packet includes acknowledgment of the new CC and data (response). In this way, T/TCP bypasses the latency of the three-way handshake.

HTTP/1.1 (RFC 2068) overcomes some of the problems of HTTP/1.0 by opening a persistent TCP connection and pipelining requests without waiting for responses. The responses are still serialized, however, which does not adequately support simultaneous rendering of inlined objects. An Internet-

Draft proposal, WebMUX, allows several protocols to multiplex so that multiple objects from a Web server can be simultaneously fetched over a single TCP Connection. In this way, metadata to objects can be sent to clients without other metadata waiting for the rest of the first object requested. HTTP/1.1 and WebMUX are session management protocols that leave unchanged the underlying transport layer, TCP.

By contrast, the idea behind *application-level framing* or ALF is that applications should be involved in the data transmission process, because applications know the requirements of the information being transmitted as well as what to do when information is lost, misordered, or delayed. Thus information should be packetized into application data units or ADUs, which should be transmitted as independent elements (instead of as a continuous byte stream as TCP assumes). The application should then determine whether ADUs should be retransmitted, discarded, or ordered.

All these proposals modify end-to-end transport protocols or attempt to bypass the byte stream-oriented semantics of TCP to better match the requirements of applications. However, they leave unchanged the IP service of point-to-point, unreliable packet transfer. This means that applications cannot be given any QoS guarantee involving end-to-end performance such as bounds on delay, delay jitter, or bandwidth.

The *Integrated Services* or IS model is a proposed extension to the Internet architecture and protocols to provide integrated services that support real-time as well as the current non-real-time service of IP (RFC 1633). The service model considers the needs of applications and those of network operators. The model provides for two categories of QoS for real-time applications. *Guaranteed service* is characterized by a perfectly reliable or worst-case upper bound on delay. *Controlled-load service* (RFC 2711) supplies a QoS comparable to a lightly loaded best-effort service. The concerns of network operators are addressed by controlled link-sharing, which guarantees a specified bandwidth to certain classes of flows.

The reference implementation framework for the IS model includes mechanisms for resource reservation, queue management, and admission control. To implement the IS model will require flows to be bound to routes, and the state of the flow to be maintained at the different routers along the route. The RSVP protocol appears to provide these mechanisms. The introduction of ATM networks in the Internet backbone provides an impetus for implementation of the IS model and the RSVP protocol (RFC 2382).

Some of the mechanisms mentioned above may be implementable by *policy-based* routing. Instead of routing solely by destination address, routers implement policies to allow or deny paths based on identity of the end system,

protocol, and size of packets. The router tags the packets by setting the IP precedence or TOS values of the IP header.

*Load balancing* is a general method to improve the speed of Web servers. The idea is to use a number of servers that contain the files. The URL requests arrive at a computer that redirects them to one of the servers. This dispatcher keeps track of the load of the servers and sends the requests to the least loaded one. This method can also be used when the servers do not all contain all the files but have instead a limited amount of replication. These parallel servers can even be made adaptive so that they can decide when it is useful to replicate information to increase the throughput of the system.

*Translators*, computers that modify the format of files, can be useful in specialized applications. For instance, consider the user of a wireless palmtop device who requests a large image from a server. Using a translator, the request goes to a translator that gets the image and compresses it before sending it to the wireless device. Many such translations are possible. For instance, e-mail can be translated into an audio clip to be accessible from a telephone; voice mail can be translated into text; video clips can be reduced to a few still images; and so on.

*Telephony servers* can be built to provide the usual telephone services for voice over IP. These services include conference calls, call forwarding, call back, as well as 800 services that reroute the calls depending on the source and on the time of day, and many others.

*Caches* are simple devices that can improve the performance of a network. When a cache is used, the Web requests are intercepted by a cache that stores files previously requested. If the cache contains the new request, then it can reply directly. If not, the cache forwards the request to the URL and copies the reply. When the cache gets full, it can use any standard replacement algorithm to make room for a new request. For instance, the cache can delete the least recently used files. Typically, cache entries have a time to live after which they are deleted. One can imagine a network of caches that exchange lists of links of which they have copies. Such a network could then be optimized to minimize the response time and the traffic on the Web.

---

## 4.7

## SUMMARY

The Internet is undergoing an explosive growth in the number of nodes, users, traffic, and applications. This growth is made possible by the simplicity of the datagram service of IP and by the hierarchical naming and addressing schemes.

However, the choice of transporting datagrams as the basic service, which may be the Internet's most important feature, may also turn out in the long run to limit its growth to applications that do not require performance guarantees in terms of delay or bandwidth. (The potential for such applications, however, is enormous as the growth of WWW suggests. The June 1998 CommerceNet/Nielsen survey found that the number of Internet users in the United States and Canada 16 years of age or older reached 79 million, up from 58 million eight months earlier.) It is easy to design new Internet protocols and routers to meet the needs of these applications. The technical challenge is to do that in such a way that the new protocols and routers can coexist with the Internet's vast legacy. If that challenge can be successfully met, the Internet could claim to provide universal networking. At the same time, advances in circuit-switched networks are enormously extending their service capability. We study those networks in the next chapter.

## 4.8

## NOTES

An interpretive account of Internet history by some of its founders is given in [LC+97]. Details about the history and growth statistics are available from the Internet Society Web site at <http://www.isoc.org/internet/history/index.html>.

Most TCP/IP documents and Internet standards are published in a series called Request for Comment or RFC. Those documents are available on-line at the URL <http://www.rfc-editor.org/>. The documents are cited as RFC ####, where #### is the RFC number.

URL is specified in RFC 1738, and a generalization of it called Universal Reference Identifier (URI) is specified in RFC 2396.

The Routing Information Protocol or RIP, based on the Bellman-Ford Algorithm is described in RFC 1058. Its extension to multicast is described in RFC 1075. Information about MBone can be obtained at <http://www.mbone.com>.

A simulation study of IP switching is reported in [LN97].

TCP/IP protocols are discussed in [T88, W98, C95]. The ideas behind TCP were first published in [CK74].

For details about the software implementations of TCP/IP in UNIX, see [C95]. The RSVP protocol is also described in [ZDES93]. VIC is available at [www-nrg.ee.lbl.gov/vic](http://www-nrg.ee.lbl.gov/vic), and VAT is available at [www-nrg.ee.lbl.gov/vat](http://www-nrg.ee.lbl.gov/vat).

ALF was introduced in [CIT90], and an implementation is described in [CKD98]. A proposal to use the ALF idea for the Web is discussed in [GCMW99].

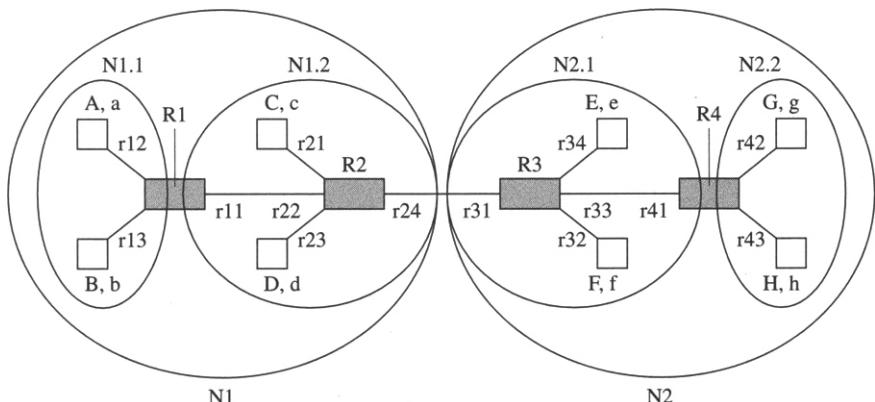
A comparison of IP-based and ATM-based service models that provide some real-time guarantees is carried out in [CWSA95]. XTP is discussed in [X92], and VMTP is described in [CW89].

The HTTP/1.1 protocol is specified in RFC 2068.

## 4.9

## PROBLEMS

1. Show that the shortest-path-first algorithm of Figure 4.8 produces a spanning tree with the shortest length, where the length of a tree is defined as the sum of the lengths of all its links.
2. Define the three-way handshake of the TCP initialization protocol in more detail, using timeouts. Explain how it avoids misunderstandings caused by a delayed packet.
3. Propose a TCP window flow-control scheme that reduces the window size when congestion is high and increases it when congestion is low. Also propose a method to measure congestion. Propose a simple mathematical model that can be used to analyze the performance of your scheme.
4. A collection of networks and subnetworks is shown in the figure below. All the links are Ethernet links. The lower case addresses are MAC addresses, and the upper case addresses are IP addresses in the net.subnet notation. For instance, r21 is the Ethernet MAC address of that interface of R2.



(a) Fill in the following tables:

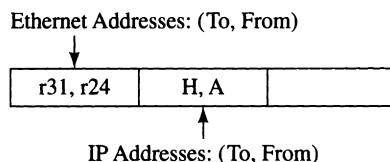
Routing table of R2:

Address	Next Hop	Port
N1.1	R1	r22
C	C	r21
D	D	r23
N2	R3	r24

ARP table of r22:

IP	MAC
R1	r11

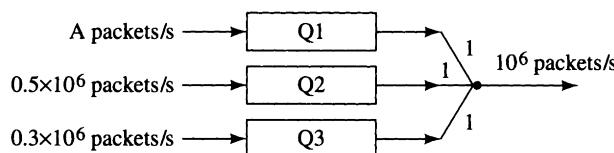
(b) Specify the addresses in the Ethernet and IP headers of a packet from A to H as it travels from R2 to R3:



#### 5. Random Early Detection (RED)

- (a) reduces the chances of successive losses of a connection.
- (b) modifies the retransmission mechanism of TCP.
- (c) modifies the window adjustment algorithm of TCP.
- (d) reduces the bias of TCP in favor of connections with a short round-trip time.

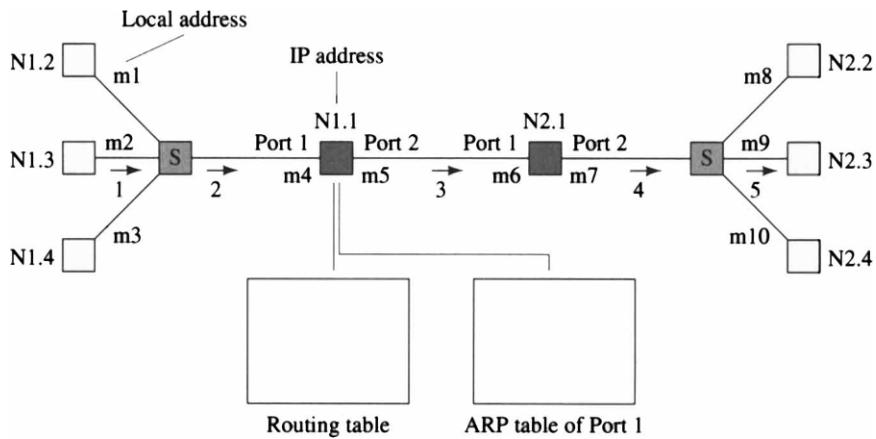
6. Consider the network below. Each switch implements a round-robin service of its three queues. That is, the switch serves one cell from queue 1, then one cell from queue 2, then one cell from queue 3, and repeats this cycle forever. It may happen that a queue overflows if it is not served fast enough. This mechanism protects streams against misbehaving connections. The switch skips empty queues.



Calculate the maximum value of A in the figure so that no cell from that stream is lost because of buffer overflow.

7. Assume that the Internet is made up of only class B networks (recall that there are up to  $2^{14}$  class B networks). We further simplify by ignoring subnetting and assuming that there is one router per class B network. For simplicity, we assume that all the local addresses are Ethernet addresses.
  - (a) What is the size (in bytes) of each entry of a router table and of an ARP table? An entry of a routing table is (Net, IP address of next router, port number). Assume that the port number takes one byte. An entry of an ARP table is (Host part of local address, Ethernet address). [Note: Round off to an integer number of bytes.]
  - (b) Estimate the maximum size (in kilobytes) of the routing table of each router and of the ARP tables of each router.
8. Imagine a perfect naming system where the root has  $N$  domains, each domain has  $N$  subdomains, each subdomain has  $N$  sub-subdomains, and each sub-subdomain has  $N$  names. Say that there are 100 million names. Estimate the number of entries of each name server table.
9. The Internet Protocol provides
  - (a) reliable packet delivery between Internet hosts.
  - (b) unreliable packet delivery between Internet hosts.
  - (c) reliable packet delivery between hosts on the same LAN or Link.
  - (d) unreliable packet delivery between hosts on the same LAN or Link.
10. Consider the network shown in Figure 4.14. To simplify, we use the class-based addressing Net.Host, with no subnetting and no CIDR.
  - (a) Fill in the routing tables and the ARP tables. A routing entry is (Net, IP address of next router, port number). An entry for a port ARP table is (IP address, local address).
  - (b) Fill in the headers of the packets 1, 2, 3, 4, and 5 for a transmission from N1.3 to N2.3.
11. The effect of congestion on TCP is to (select one or more answers)
  - (a) increase the loss rate seen by applications.
  - (b) limit the number of connections that can be set up at one time.
  - (c) reduce the throughput of connections.
12. The TCP state diagram is shown in Figure 4.15 (page 202). The states are numbered on the diagram. Consider two hosts: A and B. We indicate by  $(A_1, B_3)$  the situation when TCP is in state 1 in host A and in state 3 in host B. More generally, we use the notation  $(A_i, B_j)$ .

Assume the following scenario. Host A opens a TCP connection to host B. This start of connection is not received by host B. Host A tries again.



Headers of packets (use convention [destination | source | ...]):

1	
2	
3	
4	
5	

#### 4.14

Network addressing and routing.

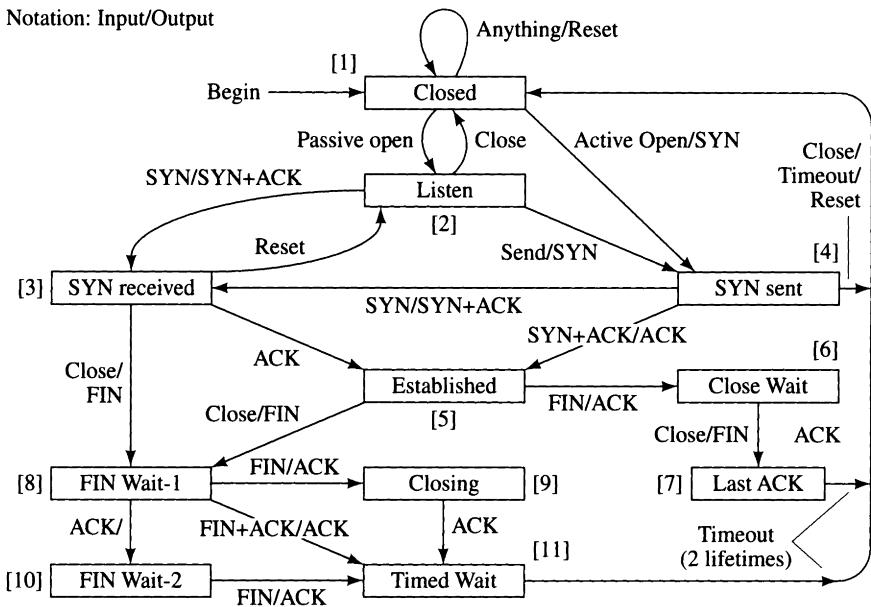
#### FIGURE

This time, host B gets the message and acknowledges. Host A then sends packets to B, who acknowledges them. Host A then closes the connection from A to B, and B acknowledges. Host B then closes the connection, and A acknowledges the close but its acknowledgment does not reach B. Host B then repeats its close message, which this time is correctly acknowledged.

Indicate the list of pairs of states of the TCP protocols in the two hosts. For instance, your answer might look like

(A1, B1), (A2, B1), (A2, B4), (A3, B4), (A3, B5).

13. Consider a router with a single FIFO queue. A random number  $X$  of TCP connections go through the router. Each connection sends bursts of packets according to Go Back N protocol. Model the queue as a discrete time queue with batch arrivals. Assume that  $X$  remains constant during a typical busy cycle of the router queue. The batch  $A(n)$  is the sum of the batches that the



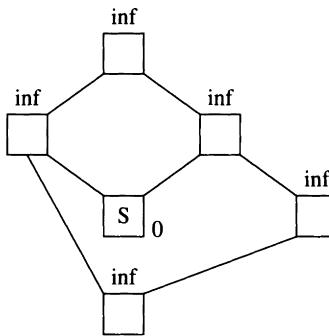
4.15

FIGURE

$X$  connections produce. At the time scale of a busy period, given  $X$ , model the  $A(n)$  as the sum of  $X$  independent, identically distributed (iid) random variables  $B(1), \dots, B(X)$  and the successive  $A(n)$  are iid given  $X$ .

- Analyze the average delay through the queue and the average queue size in terms of the distribution of  $B(1)$  and  $X$ .
  - Comment on the model and on the results of the analysis.
14. The proposals for achieving a satisfactory end-to-end quality with scalable protocols for the Internet appear to be based on the intuition that, as the number of connections increases, the relative variability of the QoS gets smaller. Using this intuition, Internet researchers propose to control QoS by using call admission control and resource allocation in the access network (e.g., controlled load, integrated services, RSVP) and to use a simple CoS in the backbone (e.g., DiffServ).  
Propose a model and analyze it to confirm or contradict this intuition.
15. Indicate the successive steps of the OSPF algorithm (Dijkstra) to find the shortest paths from the source  $S$  to all the other nodes. All the links have length 1. For each step, label the nodes, mark the nodes that should be marked, and select the appropriate links. Break ties by exploring nodes

with same smallest label in the order from left to right and from top to bottom.



16. A source sends a packet to  $N$  subscribers. For  $i = 1, \dots, N$ , subscriber  $i$  gets the packet with probability  $1 - p(i)$ , independently of one another.
  - (a) How many times must the source send the packet until every subscriber gets it?
  - (b) Now assume that the multicast occurs along a tree. The losses are independent on each branch of the tree. We can define the probability  $1 - p(i)$  that subscriber  $i$  gets the packet. However, the losses are no longer independent. Can you determine whether the number of times that the source must send the packet is larger than in the previous case or smaller?
17. What is the effect of transmission errors on the throughput of TCP? Propose a model and some analysis.

# Circuit-Switched Networks

The packet-switched networks that we studied in Chapters 3 and 4 are well suited for message traffic. Virtual circuit-switched networks, designed for variable bit rate traffic, are discussed in Chapter 6. Constant bit rate traffic is better transported by circuit-switched networks, which we study in this chapter. The most important circuit-switched networks are the telephone, cable TV or CATV, and some cellular telephone and satellite networks. We discuss only the telephone and CATV networks.

This chapter presents the underlying concepts and the technologies deployed in the 1980s and 1990s that changed the telephone system from a voice-only network to one that supports all types of traffic. You will become familiar with the most important technologies: SONET, DWDM, optical local loop, digital subscriber line solutions including ISDN and xDSL, and INA. You will also learn how CATV systems are changing from a one-way video distribution system into a network that also offers communication in the upstream direction. The telephone and CATV networks, which until 1990 served separate markets, have since begun to compete and collaborate. It is still too early to assess the full impact of the 1996 Telecommunications Bill. A notable example of that impact is AT&T's \$37 billion acquisition of the cable TV company TCI and the \$58 billion acquisition of MediaOne, with the objective of developing a two-way system that can handle voice, data, and video services. This chapter explains the technological basis underlying this change.

Today's telephone network consists of a high-speed digital backbone network. Most subscribers gain access to this network using dedicated, analog

twisted wire-pair called the *local* or *subscriber loop*. The bandwidth on these loops has been restricted to 3 kHz to make it suitable for voice. Modems that convert bit streams into voicelike signals can use this bandwidth to transmit data at rates up to 28.8 Kbps. The capacity of a channel is given by the formula  $bandwidth \times \log_2(1 + signal/noise)$ . For a 3 kHz channel with a signal/noise ratio of 30 dB or 1,000, this capacity works out to 30 Kbps. (The recent 56 Kbps modems are discussed in 5.5.2.) More than 90% of users gain Internet access through voice modems.

We introduce in section 5.1 some simple concepts that are used to describe circuit-switched networks. More elaborate mathematical models are developed in Chapters 8 and 9.

Five innovations have dramatically expanded the capabilities of telephone networks: SONET (Synchronous Optical Network), DWDM (Dense Wave-Division Multiplexer), optical local loop, various digital subscriber loop (DSL) solutions including ISDN (Integrated Services Digital Network) and ADSL (Asymmetric DSL), and INA (Intelligent Network Architecture).

SONET provides higher speeds than the current transmission system. SONET also simplifies multiplexing equipment. SONET is discussed in section 5.2. In terms of the Open Data Network model, SONET implements a "bit way," as indicated in Figure 2.28. SONET provides end-to-end fixed-rate channels at speeds that are multiples of 55.84 Mbps. Those channels can be used in a flexible manner to support a variety of bearer services. SONET is "backward-compatible" because it can carry current telephone bit streams. It is also "forward-compatible," since it can be used to transport ATM cells. The transport of ATM cells over SONET is the basis of the Broadband Integrated Services Digital Network or BISDN, which some regard as the "universal" network. There also are proposals to transport IP datagrams directly over SONET.

Five years ago, wave-division multiplexing was demonstrated in laboratories. Today commercial DWDM permits light of 40 different wavelengths, each modulating up to 10 Gbps of information, and carried on the same fiber. In this way, the capacity of existing fiber is increased 40 times at a fraction of the cost of adding new fiber. In May 1999, Nortel announced an optical amplification system that combines 160 different wavelengths, each carrying 10 Gbps of data, for a total of 1.6 terabits per second, exceeding all the traffic on the Internet. DWDM is described in section 5.3.

The third innovation in the telephone network is the partial or full replacement of the existing copper subscriber loop by optical fiber. The replacement will eliminate the current speed bottleneck intrinsic to copper, and users then will be able to send and receive all forms of information (video, image, text,

voice) requiring high speed. Only when high-speed access is widely deployed will BISDN or the “information superhighway” become a reality that many can share. In section 5.4 we study three technologies for optical local loop: the traditional digital loop carrier; passive optical networks (PON) that combine a broadcast downstream channel with a shared upstream channel; and hybrid solutions.

It is possible to remove the 3-kHz bandwidth restriction on most analog subscriber loops and make use of the 1-MHz bandwidth available on unloaded twisted pairs. The fourth innovation consists of various *digital subscriber line* or DSL solutions that use this bandwidth. ISDN, and ADSL and its variants (called xDSL), give users a combination of data and voice channels multiplexed and made available at a single interface at an aggregate bit rate of several Mbps, over the existing copper subscriber loop. DSL technologies thereby extend the economic life of the existing copper plant. We describe DSL technologies in section 5.5.

SONET, DWDM, optical local loop, and DSL technologies enhance the telephone network's information transfer services. INA is an innovation of a different sort. To appreciate INA, we must think of the telephone network as a collection of (hardware and software) resources, each of which can execute a number of activities. For example, circuits can be combined to form a path and used to transfer a constant bit rate voice stream; a recording device can be activated to record a message or to play back a previously recorded message; a switch can send a “ring” signal to a telephone; the access-line interface at the switch can record the digits dialed by a subscriber; and so on. These activities can be combined into sequences to produce valuable new services such as the 800 number service, call forwarding, voice mail, and caller identification. INA facilitates the specification and implementation of these activity sequences. INA is described in section 5.6.

The second most important circuit-switched network is CATV. Until the mid-1990s CATV was a one-way analog distribution network with a large bandwidth going “downstream” to users, little or no bandwidth going “upstream” from users, and with no switching capability. Recent technological innovations in networking and signal processing may reduce or eliminate these handicaps. If that happens, CATV will become a formidable competitor to the telephone companies because, unlike the telephone network, CATV already has a high-speed “local loop”; because it has a customer base that is comparable to that of the phone companies; and because it can use its large subscriber fees to finance the necessary investment. CATV is discussed in section 5.7.

A summary of this chapter is given in section 5.8.

---

**5.1****PERFORMANCE OF  
CIRCUIT-SWITCHED  
NETWORKS**

Circuit-switched networks consist of switches interconnected by links. The capacity or bandwidth of each link is divided into fixed-rate *circuits* (e.g., 64-Kbps circuits for the telephone network). Alternatively, we can view each link as comprising several multiplexed circuits. Telephone networks employ time-division multiplexing, while CATV, some cellular telephone, and satellite networks today employ frequency- and time-division multiplexing.

Users connect to a network switch by dedicated or shared access links. A user wishing to exchange information makes a request to the network for a *connection* or *call* to another user at a specified address. (Different networks have different addressing conventions. For example, the North American telephone network has a 10-digit address.) On receiving the call request, the network switches exchange information according to specified algorithms in order to make two related decisions: the *call admission* decision determines whether to admit or reject the request; and if the decision is to admit, the *routing* algorithm selects a route or sequence of idle circuits to connect the two users. Once the connection is established, the users exchange information. (The time taken by the switches to establish a connection is the *connection setup time*. Datagram networks have no setup time.) When the exchange is over, the user requests connection termination, and the switches return the circuits to the idle state.

A connection request that is rejected by the network is said to be *blocked*. Requests may be blocked because the switches are unable to find a route with idle circuits or because idle circuits are kept in reserve for future connection requests. The most important performance measure of the network is the *blocking probability*, defined as the chance that a call request is rejected during the busiest traffic hour.

During the information exchange period, user data is transferred in frames at a fixed rate. The frames carry a connection identifier, and the switches maintain a state table relating the connection identifier to the next link in its route. Therefore, frames in the same connection travel over the same route. Frames in different connections that go through the same link are multiplexed. At the switch, the different connections on each incoming link are demultiplexed, and different combinations are remultiplexed to form the stream for each outgoing link.

Because the network allocates a fixed bandwidth to each connection, the frames face no queuing delay. Hence the total end-to-end delay is the propagation time plus a small processing delay in each switch along the route. This total delay is both small and constant. That is why circuit-switched networks are well suited for real-time, constant bit rate traffic. In terms of the Open Data Network model of section 2.8, the bearer service offered by the network is the real-time transfer of information with specified peak rate.

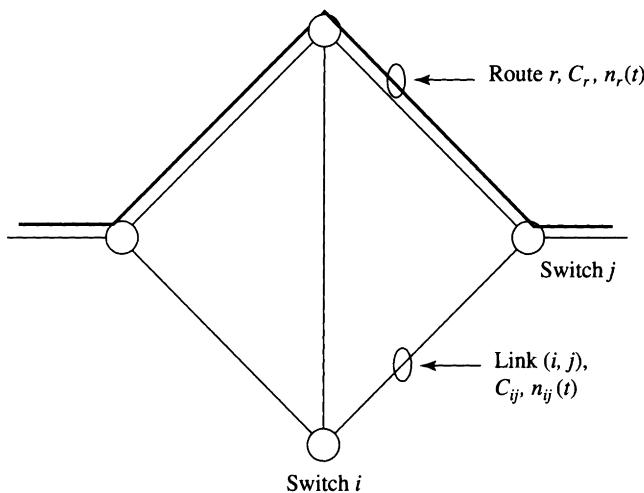
This bearer service may also be used for variable bit rate traffic whose peak rate is smaller than the connection bandwidth. This may lead to a low *utilization*, defined as the ratio of the average rate of information transfer divided by the bandwidth of the connection. If the utilization is significantly below 1, then link capacity is very poorly utilized. For example, in Telnet applications, the user's typing speed limits the information transfer to 40 bps. If this application is run over a 64-Kbps telephone line, the utilization is 40/64,000, which is less than 0.1%. The bearer service may also be used to transfer messages. In this case, the large setup time may make this use of the service noncompetitive relative to message transfer through packet-switched networks with no setup time.

Several questions of network planning, route assignment, and blocking probability can be formulated using the mathematical model of the network described next. See Figure 5.1. (We study stochastic models in Chapter 9.)

We represent a circuit-switched network as a graph whose nodes  $i = 1, \dots, n$  denote the switches, and there is an edge between nodes  $i$  and  $j$  if there is a transmission link connecting switches  $i$  and  $j$ . We suppose that the network provides connections with a fixed bandwidth. (For example, the telephone network provides 64-Kbps connections.) The capacity of the link joining  $i$  and  $j$  is denoted by  $C_{ij}$ , the number of simultaneous (two-way) connections or calls that this link can accommodate. (For example, a T-1 link can accommodate 24 voice calls; see Table 1.1.)

A route  $r$  is a sequence of connected links starting at an origin switch and ending at a destination switch. We write  $(i, j) \in r$  if link  $(i, j)$  is part of the route  $r$ . Let  $R$  denote the set of all routes. At any given time  $t$  there are a number of simultaneous active connections through different routes. We denote these connections by the vector  $n(t) = \{n_r(t), r \in R\}$ , where  $n_r(t)$  is the number of connections through route  $r$  at time  $t$ . The number of calls through link  $(i, j)$  is the number of calls through all routes that pass through  $(i, j)$ :

$$n_{ij}(t) = \sum_{\{r|(i,j) \in r\}} n_r(t).$$



**FIGURE**

Link  $(i, j)$  has capacity  $C_{ij}$  and carries  $n_{ij}(t)$  calls at time  $t$ . There are  $n_r(t)$  calls along route  $r$ , which is designed to carry  $C_r$  calls.

The number of connections through a link is limited by its capacity and so  $n(t)$  must satisfy the constraint:

$$n_{ij}(t) = \sum_{\{r|(i,j) \in r\}} n_r(t) \leq C_{ij}, \text{ for all } (i, j). \quad (5.1)$$

Relation (5.1) determines the maximum possible set of connections that can be accommodated by the network. The vector  $n(t)$  changes randomly over time. When a call along route  $r$  terminates,  $n_r(t)$  decreases by 1; if a new call is placed along route  $r$ , then  $n_r(t)$  increases by 1.

Suppose that the network is designed to carry  $C_r$  simultaneous calls along route  $r$ ,  $r \in R$ . Then the link capacities  $C_{ij}$  must be so large that

$$\sum_{\{r|(i,j) \in r\}} C_r \leq C_{ij}, \text{ for all } r \in R. \quad (5.2)$$

The system of inequalities (5.2) establishes a relation among the capacity invested in the transmission system, the choice of routes, and the maximum number of simultaneous calls that the network can accommodate.

The random amount of time that a connection endures is the call *holding* time. The random time between the arrival of two consecutive connection requests is the *interarrival* time. It is customary to denote the average holding

time by  $\mu^{-1}$  and the average interarrival time by  $\lambda^{-1}$ , so  $\lambda$  is the average rate of call requests (calls per second).

Fix a route  $r$ , and let  $\lambda_r$ ,  $\mu_r$  be the call parameter values for that route. Suppose  $p_r$  is the blocking probability for these calls. Then the rate of *carried* calls is  $(1 - p_r)\lambda_r$  calls per second. Each of these calls lasts  $\mu_r^{-1}$  seconds on average. So the average number of active connections along route  $r$  is

$$(1 - p_r) \frac{\lambda_r}{\mu_r} =: (1 - p_r)\rho_r,$$

where  $\rho_r$  is the number of call requests during one call holding time.  $\rho_r$  is an important measure of traffic intensity. Its unit of measurement is named after Erlang for his pioneering work in traffic engineering.

Evidently, we must have  $(1 - p_r)\rho_r \leq C_r$ , all  $r \in R$ , which places a lower bound on the blocking probabilities  $\{p_r\}$ . This lower bound, however, is quite optimistic, as it is based on average values and ignores the statistical fluctuations in the request arrivals and call holding times. In Chapter 9 we compute more accurate estimates of blocking probabilities.

The model described above, together with related problems presented in section 5.10, can be used as examples of how to formulate and resolve questions of planning and operations of circuit-switched networks. The parameters of costs and capacity that enter in those questions depend on technology. We now describe the main technological innovations in circuit-switched networks, beginning with SONET.

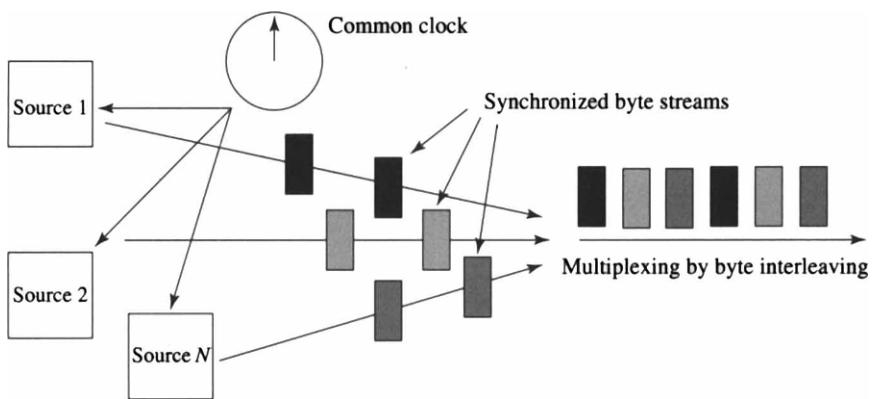
## 5.2

## SONET

We view SONET (Synchronous Optical Network) as a bit way implementation, providing end-to-end transport of bit streams. As its name suggests, the development of SONET was spurred by advances in the accuracy of clocks and in optical transmission.

SONET is an ANSI standard. (SDH is an ITU standard.) It encodes bit streams into optical signals that are propagated over optical fiber. SONET's high speed and its frame structure permit it to support a very flexible set of bearer services. The standard specifies the frame structure as well as the characteristics of the optical signal. We will discuss only the frame structure.

The most important feature of the standard is that all clocks in the network are locked to a common master clock, so that the simple time-division multiplexing (TDM) scheme of Figure 2.5 can be used. Multiplexing in SONET



**FIGURE**  
5.2

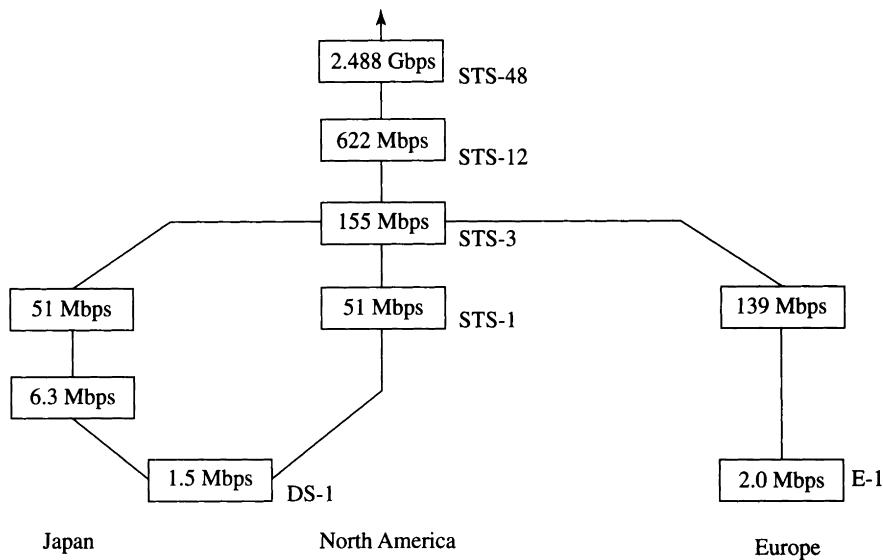
SONET sources are synchronized to a common master clock. Different streams are multiplexed by byte interleaving.

is done by byte interleaving, as depicted in Figure 5.2. As a result, as we see in the figure, if each of  $N$  input streams has the same rate  $R$ , the multiplexed stream has rate  $NR$ . Because sources are synchronized, the buffers at each input line will be very small as they have only to accommodate the effect of jitter. (*Jitter* is the inevitable, small variation in the successive clock ticks recovered from a periodic signal corrupted by some noise.)

In North America and Japan, the basic SONET signal is STS-1 (Synchronous Transport Signal-1). It has a bit rate of 51.84 Mbps. Higher rate signals are multiples of this rate; see Figure 5.3. In Europe, the basic rate is STS-3 or 155.52 Mbps, and the STS hierarchy is called the SDH or Synchronous Digital Hierarchy, starting at 155.52 Mbps.

The simplicity of the STS hierarchy makes a contrast with the current digital signal (DS) hierarchy shown in Table 1.2, which, because of the way it accommodates asynchronous streams, makes multiplexing much more complex. As Figure 5.3 suggests, moreover, SONET is backward-compatible in the sense that it can transport current telephone signals: the 1.544-Mbps DS-1 signal in North America and Japan and the 2.0-Mbps E-1 signal in Europe. SONET's frame structure is also forward-compatible in that it can support the transport of ATM cells. The frame also provides channels for organization, administration, and management in a uniform manner.

SONET's frame structure and multiplexing methods also simplify equipment. The byte-interleaved time-division multiplexing makes demultiplexing

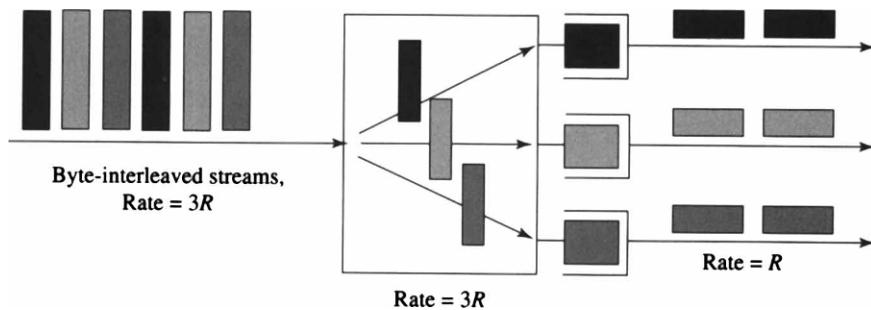


### 5.3 FIGURE

The STS- $n$  signal has a rate equal to  $n \times 51.84$  Mbps. In Europe the hierarchy starts at 155.52 Mbps. All the standards become compatible at speeds of 155 Mbps.

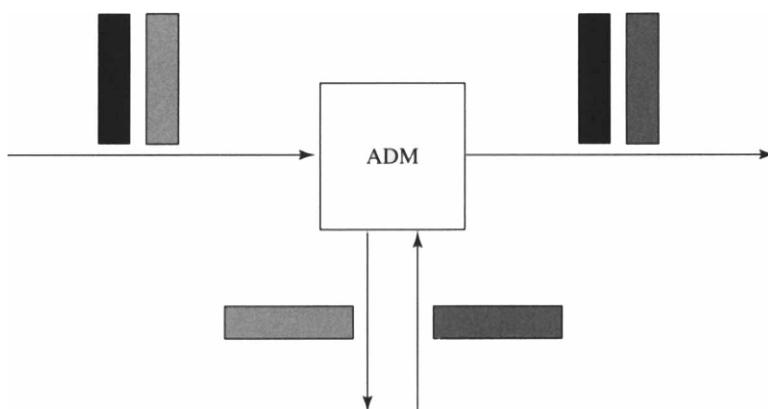
easy because, as suggested in Figure 5.4, individual streams can be determined simply by position in the frame.

A SONET ADM (add/drop multiplexer) is also less difficult to design and build. The function of an ADM is to drop one of the incoming multiplexed



### 5.4 FIGURE

SONET streams are demultiplexed byte by byte by counting.



5.5

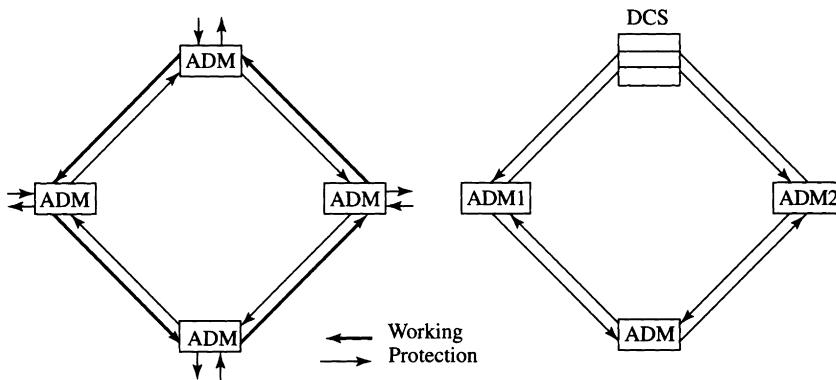
FIGURE

The add/drop multiplexer (ADM) replaces one of the incoming streams with another.

streams and replace it with another stream; see Figure 5.5. (In the current digital signal hierarchy of Table 1.2, an ADM requires a full demultiplexer-multiplexer pair.) Thus, for several functions, SONET equipment is cheaper than current equipment.

Telephone networks need to have enough redundancy and automatic procedures that use this redundancy to restore quickly the integrity of the network in the event of failures. Two examples will illustrate the magnitude of the losses that can occur when restoration procedures fail. A power outage in an unmanned central office in Hinsdale, Illinois, on Mother's Day, 1988, affected 500,000 customers who made three and a half million calls per day. Full service was not restored for one month. In November 1988, a construction crew accidentally severed a major fiber-optic cable in New Jersey, disrupting much of the long-distance traffic along the East Coast, blocking more than three million call attempts. There are many approaches to redundancy, depending on the network. We shall briefly discuss the increased reliability possible when SONET systems are deployed as dual rings. It is believed that dual rings may restore service within 50 ms of a network failure.

The basic idea is illustrated in the left panel of Figure 5.6. One fiber carries the working ring in which the signal moves counterclockwise. In the standby or protection fiber, the signal moves in a clockwise direction. Normally, there is no signal in the protection ring. If a cable between any two central offices is cut or if there is a node failure, the failure is detected by the automatic protection



5.6  
FIGURE

The panel on the left shows the configuration of a SONET ring. The panel on the right shows how digital cross-connect systems (DCSs) can be used to create logical links and reconfigure the network topology.

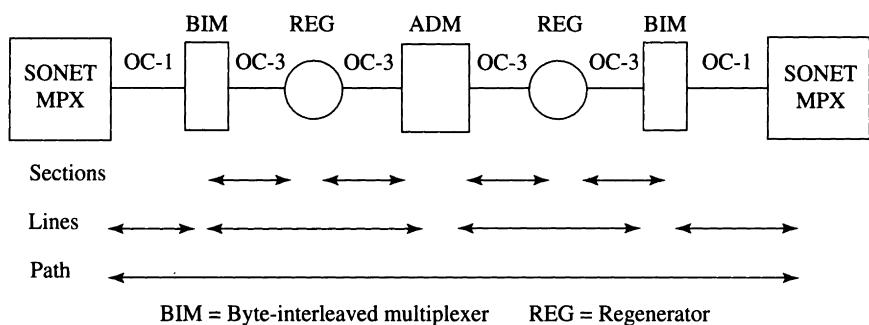
system, which performs a loop-back function using the standby fiber. (This is very similar to the FDDI recovery procedure.)

In the panel on the right in the figure, an ADM is replaced by a digital cross-connect system (DCS), which permits an incoming link to be directly connected to an outgoing link. The DCS has created a logical direct link from ADM1 to ADM2, and another logical link from ADM2 to ADM1, so that the configuration of the right panel is equivalent to a ring among the three ADMs. In this way, by setting DCS interconnections appropriately, one can effectively change the topology of the network. Such changes are carried out in order to meet shifts in demand over a 24-hour period or some other period.

### 5.2.1 SONET Frame Structure

We now describe the main features of the SONET frame structure. Figure 5.7 depicts typical transmission equipment connecting a network element (on the left) where a SONET frame is assembled to another element (on the right) where that frame is disassembled.

(Two terms in the figure may be obscure. A *regenerator* (REG) is a device that boosts the power of the optical signal. The device has three components: the first component converts the optical signal into an electrical signal, the second amplifies the electrical signal, and the third converts the amplified electrical signal back into a more powerful optical signal. The second possibly obscure



## 5.7

The main SONET network elements.

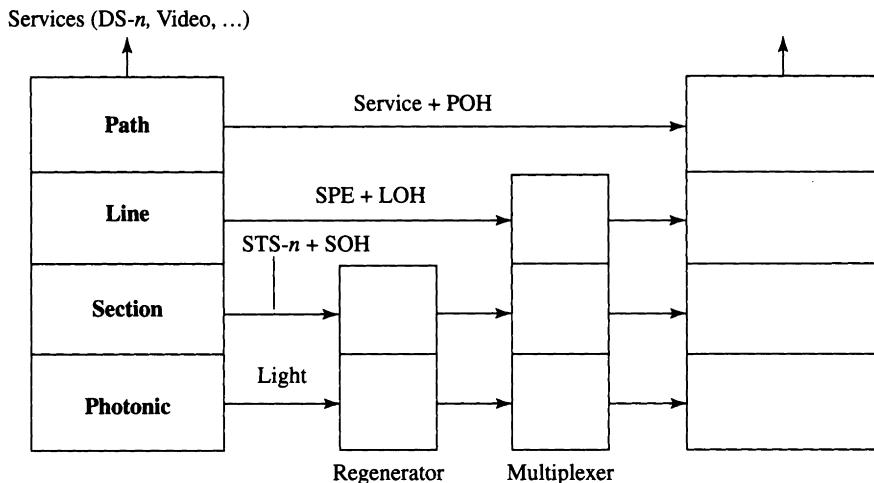
FIGURE

term is *OC-n* or *Optical Carrier-n*. It is the name of the facility that transmits the STS-*n* signal.)

Figure 5.7 also gives certain definitions used in the SONET frame overhead structure. *Section* is the portion of the transmission facility between a terminal network element and a regenerator or between two regenerators. A *terminal point* is the point after signal regeneration at which performance may be monitored. *Line* is the transmission medium (optical fiber) and associated equipment required to transport information between two consecutive network elements, one of which originates the line signal and the other of which terminates it. (Line corresponds to what we call a link.) A line has a certain bit rate. *Path* at a given rate is a logical connection between a point at which a standard frame for the signal is assembled and a point at which the standard frame is disassembled. (Path corresponds to an end-to-end route.)

The layered overhead structure associated with these definitions is given in Figure 5.8. The four layers of the structure are the path, line, section, and photonic layers. Their main functions are summarized in Table 5.1.

Consider an example where two path layer processes are exchanging DS-3 frames. (The 45-Mbps DS-3 signal belongs to the current telephone signal hierarchy.) The DS-3 frames, plus POH (path overhead), are mapped into an STS-1 SPE (synchronous payload envelope), which is then given to the line layer. The line layer may multiplex several different payloads (in addition to the STS-1 SPE containing the DS-3 signal) from the path layer (frame and frequency aligning each one) and add LOH (line overhead). In addition to multiplexing, the line overhead provides other functions, such as protection switching. Finally, the SOH (section overhead) provides framing and scrambling prior to transmission by the photonic layer. The photonic layer converts the electrical bit stream from the section layer into an optical signal.



5.8

The four layers of the SONET overhead.

FIGURE

Going bottom up, each layer builds on the service provided by the layer below. The photonic layer provides optical transmission at some rate; the section layer provides framing and scrambling for the bits being transmitted; the line layer provides line maintenance and protection and multiplexing of STS-1 signals; the path layer provides a mapping from services (e.g., DS-3 frames, ATM cells, digital voice streams) into STS-1 SPEs. In terms of the Open Data Network model, the SONET path layer implements the bearer service consisting of end-to-end transport of bit streams in the format of the SPE.

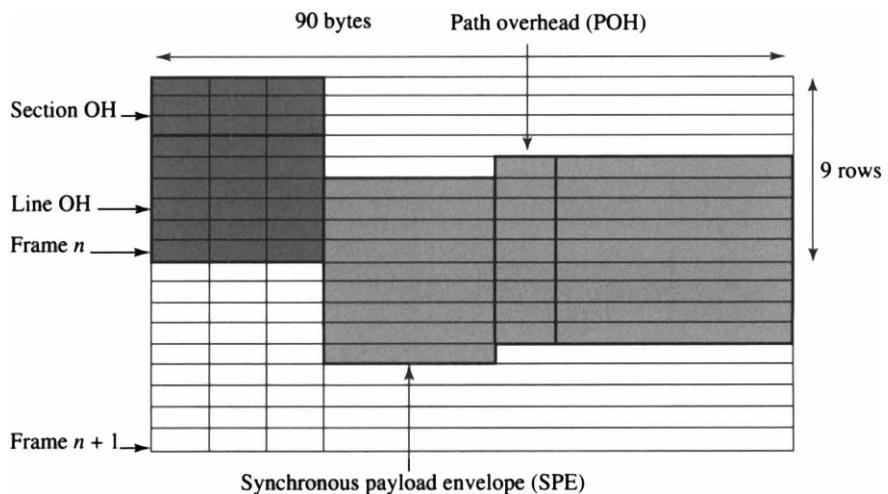
Certain network equipment may not participate in some of the upper-layer functions. For example, the multiplexer in Figure 5.8 operates at the line layer,

Layer	Function
Path	Services; end-to-end error detection
Line	Multiplexing, with frame and frequency alignment; protection switching; data links
Section	Framing, scrambling, data links
Photonic	Electrical to optical and optical to electrical signal conversion

5.1

The layers of SONET and some of their functions.

TABLE



5.9

FIGURE

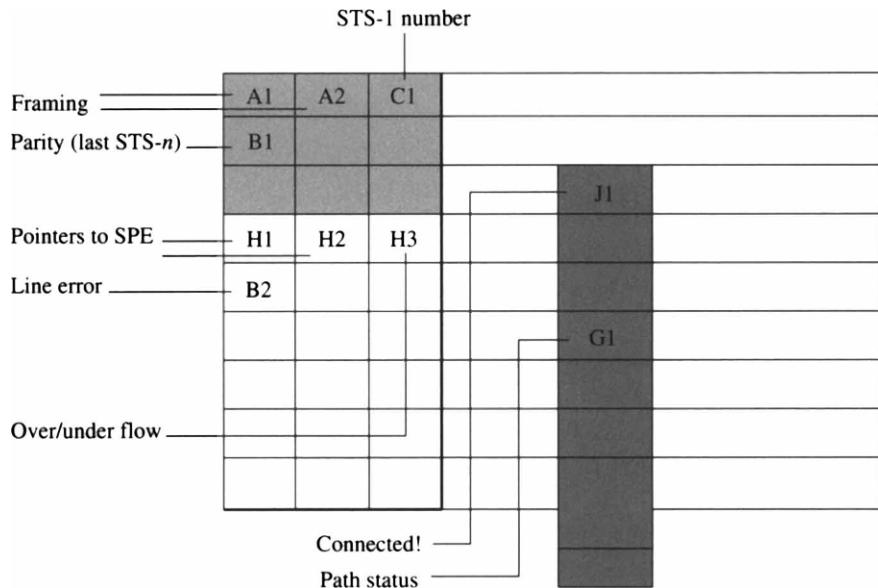
and the regenerator operates at the section layer. A photonic amplifier (which directly amplifies the optical signal without converting it into an electrical signal as a regenerator does) operates at the photonic layer.

Figure 5.9 presents overall features of the SONET frame. The 810-byte STS-1 frame is organized into nine rows of 90 bytes (transmitted left to right, top to bottom). A frame is  $125 \mu\text{s}$  in duration, corresponding to one 8-kHz voice sample. (This gives the STS-1 rate of  $810 \text{ bytes/frame} \times 8,000 \text{ frames/s} = 51.840 \text{ Mbps}$ .) The first three columns (27 bytes) are for SOH and LOH. Thus an SPE is  $9 \text{ rows} \times 87 \text{ columns}$ , of which the first column (9 bytes) is devoted to POH.

The most unusual feature is that an SPE does not need to be aligned to a single STS-1 frame: it may “float” and occupy parts of two consecutive frames as shown in the figure. Two bytes in LOH are allocated to a pointer that indicates the offset in bytes between the pointer and the first byte of the SPE.

Figure 5.10 explains the function of some individual overhead bytes. The standard defines the function of each byte and assigns it to a layer and position in the frame. Each of most of the unidentified bytes provides a 64-Kbps data link for use by the transmission equipment for sending messages, commands, and alarms for transmission functions. Examples of these functions are testing, protection switching, and fault isolation.

The first three rows are assigned to SOH. Two bytes (called A1, A2 in the standard) are for framing each STS-1. Another byte (C1) is the STS-1 channel ID. It is a unique number assigned to each channel prior to byte interleaving.



5.10

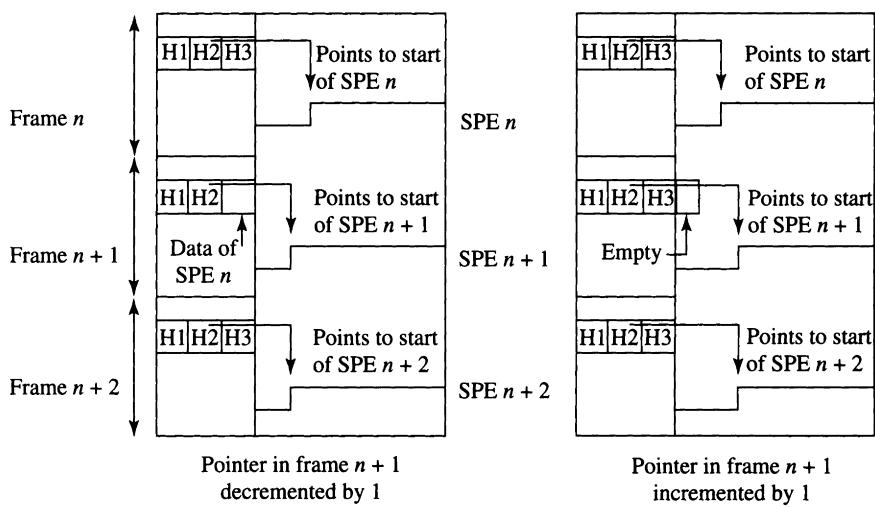
Function of some overhead bytes.

FIGURE

into an STS-*n* signal and stays with that channel until it is deinterleaved (demultiplexed). It is used to determine the position of the other STS-1 signals. Byte B1 is a bit-interleaved parity code 8 (BIP-8) using even parity. The section BIP-8 is calculated over all bits of the previous STS-*n* after scrambling. It is defined only for the first STS-1 tributary of an STS-*n* signal.

The remaining six rows are for LOH. Byte B2 is another parity code used for line-error monitoring. Bytes H1, H2 are allocated to a pointer indicating the offset in bytes between the pointer and the first byte of the STS SPE. Byte H3 is used for frequency justification, that is, to compensate for clock deviations. Even though all sources are synchronized to the same master clock, deviations do occur. When the rate of the source (payload) is higher than the local STS-1 rate, H3 is used to add an extra byte to the SPE; when the source is slower, a byte is deleted from the SPE. We explain how this works in detail using Figure 5.11.

In any frame (H1, H2) points to the start of the SPE in that frame. Suppose that the payload speed is higher than the frame speed. Then, on occasion, one extra byte is added to the payload. This is done by decrementing the pointer (in frame *n* + 1) by 1, so that SPE *n* + 2 starts sooner. The extra byte is placed in H3.



5.11  
FIGURE

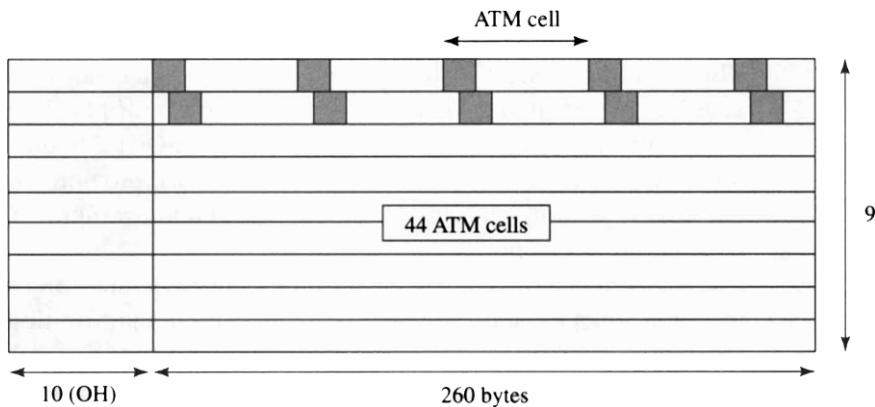
Frequency justification: when the payload rate is higher than the frame rate, H3 is used to provide an extra byte as on the left; when the payload rate is lower, a byte is left empty or “stuffed” as on the right.

Suppose instead that the payload bit rate is slower than the frame rate. Then, on occasion 1 byte is not made available to the payload. This is done by incrementing the pointer (in frame *n + 1*) by 1, so that SPE *n + 2* starts later. The byte next to H3 is now left empty or “stuffed” and not allowed to be part of the payload envelope.

The POH is assigned and remains with the payload until the payload is demultiplexed. The first byte of POH, J1, is used to repeat a (user-programmable) 64-byte, fixed-length string so that a path-receiving terminal can verify its continued connection to the intended transmitter. The third byte, G1, is used to convey back to an originating STS path terminal the path status and performance. This feature permits the status and performance of the complete duplex path to be monitored at either end or at any point along that path.

Figure 5.12 illustrates how an STS-3 frame would be used to carry (approximately) 44 ATM cells of 53 bytes each. An interesting feature is that no framing bits are provided to delimit the boundary of the ATM cells within the payload. We will see in Chapter 6 how the CRC bits of the ATM cell header are used to find the cell boundary.

In summary, SONET is a transmission standard that assumes a synchronous mode (all signals synchronized to the same clock). Its flexible frame

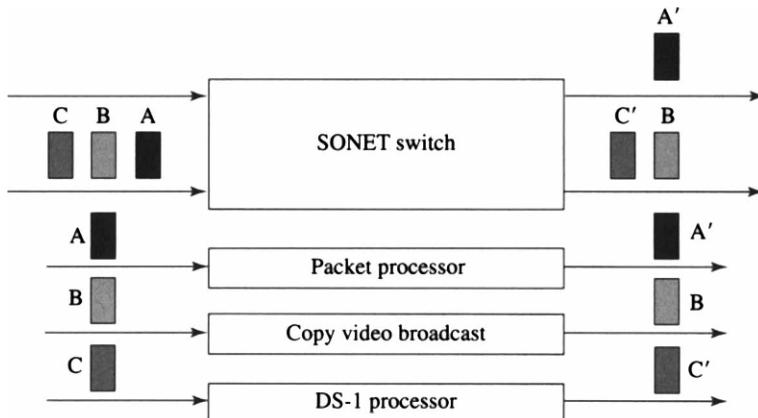


5.12

**FIGURE**

An STS-3 frame accommodates 44 ATM cells. No framing bits are provided to delimit the cell boundary.

structure accommodates both asynchronous traffic such as ATM and synchronous transfer mode (STM) traffic such as voice. These two kinds of traffic can be separated and recombined at an add/drop multiplexer or a switch, as suggested in Figure 5.13. Whether a particular service will follow an STM or ATM mode will depend on the traffic characteristics.



5.13

**FIGURE**

At a SONET switch, streams of different traffic types are demultiplexed, appropriately processed, and then remultiplexed.

### *Optical Networking and Future of SONET*

The synchronous clock in the nodes of a SONET network facilitates byte interleaving, reducing the cost of time-division multiplexing equipment. That feature encouraged the widespread adoption of SONET in telephone networks. Motivated by the needs of telephony, SONET also provides signaling channels for fault detection, automatic protection, and management. SONET redundant ring architectures provide reliability.

High-speed packet-switching routers and switches combine packets from multiple links by statistical multiplexing rather than by time-division multiplexing, and so the signals on those links need not be synchronized. (Nor is there a need for the still-expensive multiplexers and circuit switches.) At the physical layer, high-speed packet-switched networks may use optical fiber terminated by SONET line interface cards including the features that promote reliability, but may dispense with the central clock and the features for frequency justification needed for time-division multiplexing. These "lightweight" SONET links are used in Gigabit Ethernets and ATM. Efforts are also underway to place IP packets directly into SONET frames (bypassing the ATM layer). In this approach, IP datagrams are encapsulated into Point-to-Point Protocol (PPP) packets, which are then framed using HDLC to delineate the PPP packet within a SONET payload envelope.

As the QoS guarantees of ATM become well established, ATM will carry both voice and data. At that point, the flexibility and efficiency of statistical multiplexing, combined with the lower cost of ATM switches compared with TDM circuit switches and multiplexers, will bring an end to the growth in the use of circuit switching for the backbone telephone and data networks. Thus the future backbone networks (for both data and voice) are likely to deploy ATM or IP switches and routers over "lightweight" SONET links (not SONET networks). SONET links and interfaces will also be used as the physical layer in high-speed LANs.

Thus ironically, although it was the synchronous nature of SONET networks that encouraged large-scale SONET deployment and reduced the cost of SONET line interfaces, it is the latter that will survive in networks that are asynchronous. ATM and IP networks over SONET links will account for an increasingly large fraction of future growth for one further additional reason: it is much easier to interconnect packet-switched networks than to interconnect TDM circuit-switched networks. By interconnecting with existing networks, a small ATM or IP network company can effectively compete with a much larger TDM-based telephone company.

Future high-speed networks will evolve from TDM circuit-switched infrastructure into packet technology that combines high-performance cell-switching and routing, with statistical multiplexing, initially over SONET links. Over time SONET functions will be moved into the data equipment.

## 5.3

### DENSE WAVE-DIVISION MULTIPLEXING (DWDM)

An optical link consists of a laser that emits a light of a certain wavelength, modulated by a data stream. The modulated lightwave is sent down an optical fiber. At the other end of the fiber, the received lightwave is demodulated, and the data stream is recovered. The optical fiber has a bandwidth of about  $25 \times 10^{12}$  Hz, as we explain in Chapter 11. However, electronic components today are limited to modulation speeds of 2.5 Gbps, so most of the fiber bandwidth is unused.

In wave-division multiplexing, lasers of different wavelengths are modulated by separate data streams, and the modulated lightwaves are passively combined (added up) and sent down the same fiber. At the other end the different wavelengths are separated out, individually demodulated, and all the data streams are recovered. Thus wave-division multiplexing creates several different channels over the same fiber, each with a bit rate of up to 2.5 Gbps (SONET OC-48), expected to increase to 10 Gbps (OC-192).

Commercial dense wave-division multiplexers (DWDMs) combining up to 16 wavelengths were introduced in 1996; 40-channel availability was announced in 1998; a 160-channel commercial system, each with a 10 Gbps data rate, was announced by Nortel in 1999. Since DWDM equipment can be used with existing fibers, the capacity of the links in the telephone and Internet backbones has dramatically and suddenly increased from 2.5 Gbps to 100 Gbps. Most DWDM equipment is designed to work with SONET, but products have been announced with an “open configuration” that can also transport ATM cells or IP packets, without an intervening synchronization layer like SONET.

As we will see, however, full utilization of this capacity will require innovations in switching technology that can incorporate DWDM links into an optical network.

The success of DWDMs is due to the development of the distributed feedback laser structure that provides a stable, monochromatic light source, and filters that separate lightwaves spaced 100 GHz apart. The development of

wideband optical amplifiers that boost the power of all multiplexed lightwaves simultaneously (without optical/electrical conversion) has increased the span of optical links, further reducing cost and increasing reliability.

When DWDM links form a network, as in Figure 2.1, a switching element is needed to transfer the bit streams modulating the light signal from an incoming link to an outgoing link. Today, these switches are digital. The incoming multiplexed signal is demultiplexed into component lightwaves, each of which is demodulated to recover the individual bit streams. The bit streams are electronically switched, individual lasers are modulated, and the lightwaves are remultiplexed into the fibers at the output ports. Several efforts are underway to bypass this costly digital switching, by performing the operations purely in the optical domain. Without the creation of such optical switching elements, the use of DWDM links will be limited to long-haul point-to-point links.

The first commercial optical add/drop multiplexers have been announced. These elements can add or drop a small number of individual wavelengths at a single site without demultiplexing the entire wave-division multiplexed signal. These add/drop multiplexers can be used at the network termination nodes and will promote the use of DWDM in metropolitan areas.

Wavelength-selective optical cross connects (WSXC or OXC) have already been demonstrated. These are switches with a certain number of input and output ports. Each input port accepts a signal multiplexing say 8 wavelengths. The different wavelengths are demultiplexed at the input port, switched through the fabric to arrive at different output ports, where they are remultiplexed. Thus a signal arriving at one input port on one wavelength is transmitted essentially unchanged from an output port, after power loss compensation. In this way a virtual link can be created comprising a single wavelength routed through several WSXCs. This is called *wavelength routing*. The limitation of this approach is that two signals modulating the same wavelength arriving on different ports cannot be transmitted on the same output port.

The limitation of wavelength routing is that the same wavelength must be used in the virtual link. This limitation could be overcome by the development of *wavelength converters* that convert an input wavelength into a different wavelength while being transparent to the modulating data. Changing the configuration of cross connects is still a slow process. It is used only to track large, predictable changes in the aggregate pattern of traffic. So at this time this technology is not suitable to dynamically reconfigure virtual links at the time scale of individual connections.

Further in the future are optical switches that can switch individual ATM cells carried by a lightwave at one input port to a lightwave at another port.

Such devices must perform cell header processing and cell buffering in the optical domain, in addition to fast switching.

WDM technology is further discussed in section 11.2.

## 5.4

## FIBER TO THE HOME

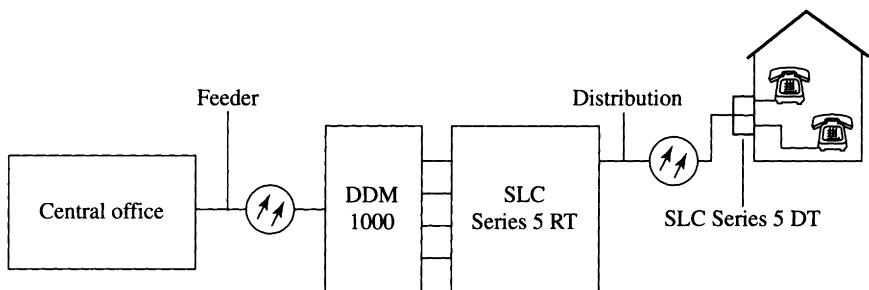
In this section we consider approaches designed to address the “last mile” problem: Assuming that a high-speed backbone network is available, how can residential or business customers be connected to it in a cost-effective manner? The approaches considered here rely on replacement of the copper subscriber loops by optical systems. These systems are known as *fiber in the loop* (FTFL) or *fiber to the home* (FTTH). CATV systems are sometimes also grouped under this heading.

There is an economic justification for using fibers in the subscriber loop when fibers are less expensive than copper or when new services require a larger bandwidth than that provided by copper. The cost of installing a fiber loop is comparable to that of a copper loop. As a result, local operating companies often deploy optical loops for new installations. Some are replacing old copper loops with fibers when maintenance is needed or anticipated. The use of fibers does entail additional cost of electric to optical conversion.

Most observers believe that the installation of fiber will be justified by the anticipated new services that require high-speed access. Examples of these services are ISDN, TV distribution, home shopping, and high-speed Internet access. Working at home (telecommuting) and distance learning may contribute greatly to the demand for high-bandwidth services. However, increased access speeds over the existing copper loops and CATV will retard the deployment of optical loops.

### 5.4.1 The Optical Loop Carrier System

Most telephones are connected to the central office by a twisted wire-pair. In the 1970s, T-1 carrier lines were introduced into loop cables, terminated by *digital loop carrier* systems, one in the central office and the other in a remote terminal (RT). Subscriber signals, carried over twisted pairs, were time-division multiplexed at the RT into the T-1 carrier to and from the central office. In the early 1990s, optical fiber transmission began to be installed between the central office and the RTs in place of the T-1 lines.



RT = Remote terminal, DT = Distant terminal, DDM 1000 = Digital multiplexer

**5.14**

**FIGURE**

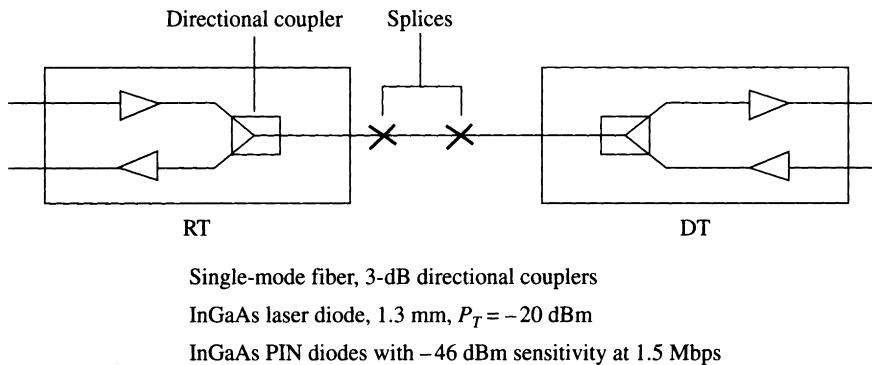
The time-division multiplexer at the RT can be software-configured to provision multiples of 64 Kbps service to individual subscribers, as needed. New carrier systems offer more flexible subscriber access including Frame Relay and ATM. In addition to this flexibility, the loop carrier systems eliminate the cost of the line interface cards needed per subscriber wire-pair terminating in the central office.

We now describe one carrier system in more detail. Figure 5.14 gives a block diagram of the AT&T Subscriber Loop Carrier (SLC) Series 5 system. In this system, the signals to and from different subscribers are multiplexed electronically (TDM) at a remote terminal (RT). Fibers instead of wire-pairs connect the remote terminal to the customer premises. Thus the optical subscriber signal is converted to an electric signal at the RT, the different electric signals are multiplexed, and the multiplexed signal is converted to an optical signal and sent to the central office. The characteristics of the optical subscriber loops are indicated in Figure 5.15. (These characteristics may not seem meaningful. In Chapter 11 we study what they mean.)

The system allocates a DS-1 (1.544-Mbps) channel to each customer location. As a result, additional lines and ISDN services are easily provided to the customers.

### 5.4.2 Passive Optical Networks (PONs)

The loop carrier systems deliver a fixed bit rate, dedicated, duplex channel to each subscriber using TDM. The passive optical networks or PONs described in



5.15

#### Details of the AT&T subscriber loop system.

## FIGURE

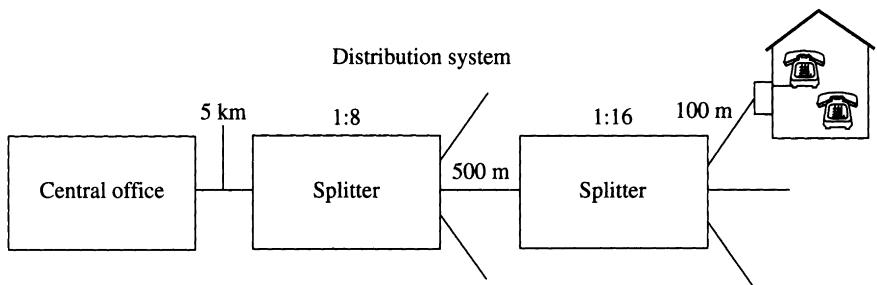
this section consist of a broadcast downstream channel and a shared upstream channel. The downstream channel multiplexes several data streams using TDM, and subscriber terminals select the stream intended for them. In the upstream channel, data streams from individual subscribers are simply added together using passive optical couplers. In order to prevent overlapping of different data streams, each subscriber stream is allocated a fixed time slot. A MAC protocol governs this time-slot allocation.

Cable modem systems for CATV use a similar approach. The important difference is that in PONs, the broadcast and multiplexing functions are carried out passively in the optical domain.

We describe in detail the British Telecom TPON (Telephony on PON) system, and then briefly discuss more recent proposals.

Figure 5.16 depicts the TPON system. This system uses passive (optical) multiplexing to reduce the cost per customer. The TPON topology is a “double star” that attaches 128 ( $16 \times 8$ ) customers on each fiber from the central office. The two splitters in the figure are optical (contrast with Figure 5.14, where multiplexing and demultiplexing are done electronically, requiring electrical/optical conversion).

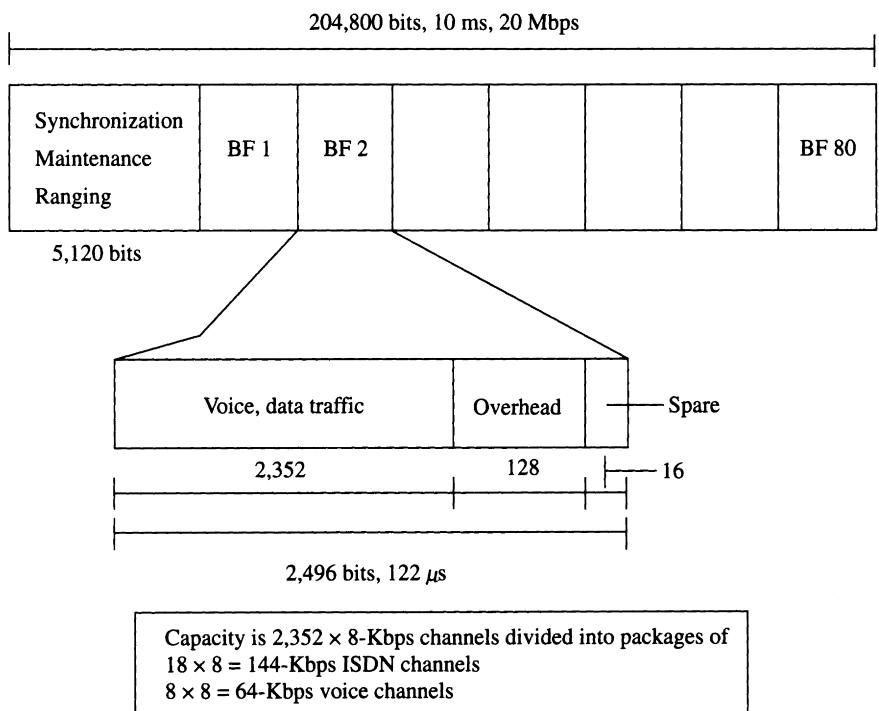
Figure 5.17 shows the frame structure of TPON for the downstream signal from the central office to the customers. The frames are repeated 100 times per second, and each frame contains 80 slots. Thus, slots are repeated 8,000 times per second, so that each byte (8 bits) repeated in all the slots constitutes a 64-Kbps channel, which may be used for voice or data. Each 2,496-bit slot thus provides 2,352 8-Kbps channels for 128 users plus overhead channels. These channels can be packaged as voice channels, ISDN services, data channels, or



5.16

FIGURE

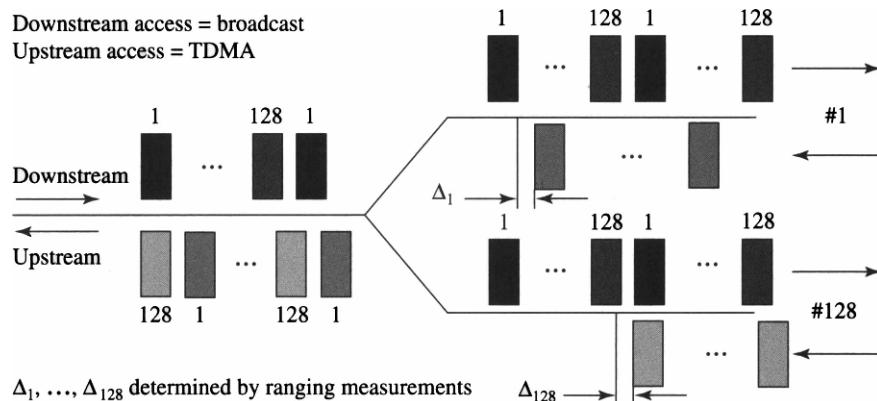
The British Telecom TPON system.



5.17

FIGURE

The TPON frame structure.



5.18

Streams are multiplexed, without overlapping.

FIGURE

their combinations. The system uses the additional bits in each slot and each frame for synchronization and maintenance. Some of these bits are also used for ranging measurements needed for multiplexing of the signals coming from the customers. As shown in Figure 5.18, the signals from the customers are multiplexed in time. The 128 subscriber units time their periodic transmissions so that their signals are interleaved without overlapping when they reach the center of the star (the central office). (Signal overlap is analogous to a collision on the Ethernet.) To reduce the chance of overlap, a “guard band” must be provided around each individual transmission, which wastes potential bandwidth.

TPON services offer dedicated bandwidth to subscribers using TDM, similarly to loop carrier systems and DSL. The economic viability of TPON has encouraged more ambitious European proposals.

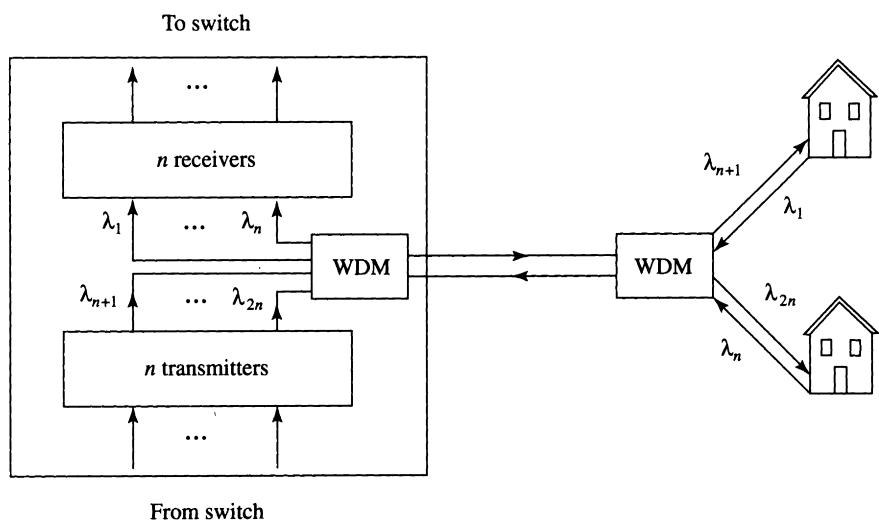
APON, or *ATM over PON*, systems attempt to combine the advantages of ATM-based statistical multiplexing and the broadcast nature of PON, to provide much higher bit rates. (A recent proposal is for 600 Mbps downstream and 300 Mbps upstream.)

The physical double-star layout of APON is similar to TPON. However, the use of optical amplifiers permits a greater splitting ratio and longer distances. The frame structure is also similar to TPON. Fixed-size frames are divided into slots allocated to ATM cells plus overhead. Subscriber units periodically conduct ranging measurements and a two-bit guard band is provided to prevent overlap of signals from different subscribers.

One new issue must be addressed in order to accommodate statistical multiplexing. Since subscribers are not given fixed slots in which to transmit, a MAC protocol is needed that allows subscribers to request and be allocated slots in which to transmit. Reservation bits in upstream slots are set by subscribers to indicate their requests. Downstream slots carry grants to individual subscribers. Grants and upstream requests are synchronized in such a way that in the upstream direction a near-continuous stream of cells is multiplexed on the fiber.

### 5.4.3 Passive Photonic Loop (PPL)

The passive photonic loop (PPL) uses wave-division multiplexing or WDM instead of TDM. PPL allocates one pair of wavelengths of light to each subscriber. Thus  $n$  subscribers need  $2n$  different wavelengths. A subscriber transmits using one wavelength and receives the other wavelength. A block diagram of the PPL system is shown in Figure 5.19. In the figure, subscriber 1 is allocated wavelengths  $\lambda_1, \lambda_{n+1}$ , subscriber 2 is allocated wavelengths  $\lambda_2, \lambda_{n+2}$ , and so on. The subscriber can modulate light of one wavelength to send information to others;



5.19

FIGURE

The PPL architecture allocates one pair of wavelengths of light to each subscriber.

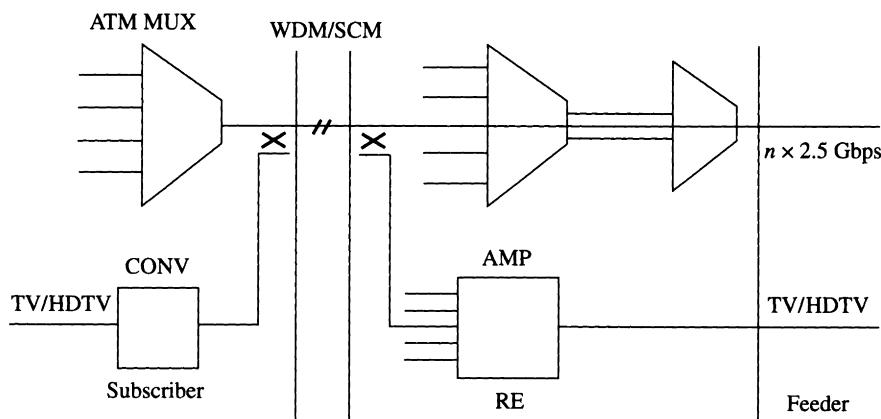
and others can modulate light of the second wavelength to send information to the subscriber. There is a modulation bandwidth around each wavelength, which defines the maximum transmission speed. In this sense, WDM is similar to frequency-division multiplexing (FDM), described in section 2.6.1. The bandwidth available to each subscriber is huge, on the order of hundreds of GHz, although current technology limits utilization to a few Gbps. (The technology is discussed in Chapter 11.)

The number of subscribers that PPL can accommodate is limited by the number of wavelengths that can reliably be differentiated using tunable lasers and optical filters. Current technology places an upper bound of 50.

PPL networks today exist only as laboratory experiments. Many advances are needed before such networks will be commercially viable.

#### 5.4.4 Hybrid Scheme

Hybrid schemes that combine time-division and frequency- or wave-division multiplexing are also possible. One such system is shown in Figure 5.20. In this system, a number of bit streams are transported over ATM, using time-division and statistical multiplexing. The resulting ATM cell streams are multiplexed with some TV signals either by subcarrier multiplexing (described in Chapter 11) or by WDM. Such a system can be used to superpose CATV and ATM networks on the same fiber system. Since the systems can make



5.20

A hybrid scheme.

use of parts of existing distribution systems for CATV and telephone, these systems may prove to be very cost-effective to deploy in the next stage of service integration.

## **5.5**      **DIGITAL SUBSCRIBER LINE (DSL)**

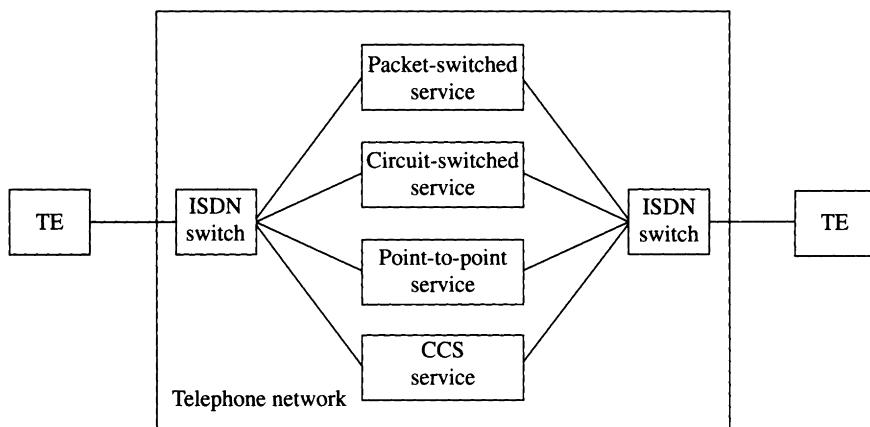
If the 3-kHz restriction placed on the twisted pair connecting a subscriber's telephone and the central office is removed, it is possible to use the 1-MHz bandwidth that is available for transmitting data. *Digital subscriber line* or DSL is the generic term for a set of technologies that use this bandwidth for digital transmission. The term was first used for basic rate integrated services digital network or ISDN, which provides 128 Kbps access.

In the early 1990s, telephone companies initiated efforts to use the 1-MHz bandwidth for transmission of video over twisted pairs. The service needed greater bandwidth in the downstream direction. The efforts culminated in the development of the *asymmetric digital subscriber line* or ADSL. The video distribution application turned out not to be commercially viable. However, a growing demand for ADSL products began in 1997. Today, high-speed Internet access is seen as the main application of ADSL, especially for the so-called SOHO (single office, home office) market. Analog voice-modems offer 28.8 Kbps. ISDN increases this to 128 Kbps, compared with ADSL speeds of between 1.5 Mbps and 8.0 Mbps. Unlike ISDN, ADSL standards are not widely accepted, and there are several competing schemes, collectively known as xDSL.

### **5.5.1**      **ISDN**

Telephone companies are implementing the Integrated Services Digital Network (ISDN) according to standards defined in successive recommendations culminating in [I120]. The objective of ISDN is to offer new digital transmission services to subscribers. The telephone network uses digital transmission for voice and a packet-switched X.25 network for the transport of signaling information. ISDN makes these internal transport facilities available to users as new services.

ISDN offers a variety of bearer services built on top of the first three OSI layers, higher-level services (called *teleservices*), and supplementary services. The teleservices are application-layer services in terms of the Open Data Network model. The supplementary services are concerned with call control



5.21

FIGURE

ISDN architecture. Circuit, packet, unswitched point-to-point, and call-control services are accessed through a common interface.

functions rather than communication per se, and do not fit directly in the Open Data Network model.

The main bearer services are the transport of audio and digitized voice (at 64 Kbps), circuit-switched digital channels at rates that are multiples of 64 Kbps, packet-switched virtual circuits, and connectionless service (datagrams). The teleservices include telex, facsimile, videotex, and teletex transmissions with specific coding and end-to-end protocols. A message-handling service and a directory service are also being defined for ISDN. The supplementary services include telephone services such as caller identification, call forwarding, call waiting, and conference calling. Figure 5.21 gives a schematic of the ISDN architecture. Circuit-switched, packet-switched, dedicated point-to-point, and call-control (common channel signaling or CCS) services are brought together at an ISDN switch and accessed by the user through a common terminal equipment (TE).

The user interfaces to ISDN are defined as combinations of three types of channels: B, D, and H. (See Figure 5.22.) The B channel is a 64-Kbps channel that transports a circuit-switched connection, an X.25 service (packet-switched, virtual circuit), or a permanent digital point-to-point connection. The D channel is a 16-Kbps or a 64-Kbps channel used for signaling information (call control) and for low bit rate packet-switched services. An H channel is a 384-Kbps, 1,536-Kbps, or 1,920-Kbps channel used like a B channel but for higher-rate services. The ISDN standards specify the basic access and the

Channel types:	Basic interface:
B: 64 Kbps CS, X.25 (PS, VC), or permanent signaling and low rate PS	Pseudo-ternary ( $1 = 0 \text{ V}$ , $0 = \pm 0.75 \text{ V}$ alt.)
D: 16 Kbps or 64 Kbps	Frame level: 192 Kbps, sync., DC balancing (144 Kbps)
H: 384 Kbps, 1,536 Kbps, or 1,920 Kbps as B	Data link:
Basic access:	B/PS: LAPB (GBN, ACK + NACK) B/CS and Permanent: user's choice
2B + D (16)	D: LAPD: acknowledged = GBN, VC unacknowledged: datagram with discard
Primary access:	Network: routing, mpx, congestion control, call control
30B + D (64) in Europe 23B + D (64) United States, Japan, Canada	

---

**5.22****ISDN services and standards.****FIGURE**

primary access for users. The basic access is  $2B + D$ ; it consists of two full-duplex B channels and a full-duplex 16-Kbps D channel. The primary access is  $30B + D$  (64 Kbps) in Europe, and it is  $23B + D$  (64 Kbps) in the United States, Japan, and Canada.

The ISDN standards specify a network-user interface that can be accessed directly by ISDN terminal equipment such as digital telephones and, via terminal adapters or ISDN routers, by other devices such as computers.

We now summarize some of the ISDN standards for implementing the lower three OSI layers. For basic access, the physical layer of ISDN specifies an 8-pin connector to attach to the network, a pseudo-ternary encoding (1 is represented by 0 volt and 0 by alternatively +0.75 volt and -0.75 volt), a frame format that includes synchronization and DC-level balancing bits, and a line rate of 192 Kbps corresponding to the 144 Kbps of user data rate ( $2 \times 64 + 16$  Kbps) plus the overhead bits. In addition, the physical layer specifies a contention-resolution protocol for access to the D channel by up to eight terminals attached to a common (multidrop) line. (See Figure 5.22.)

The data link layer of ISDN is LAPD for the D channel and LAPB for packet-switched connections on the B channel. For circuit-switched or permanent connections on the B channel, the users can choose the data link protocol and can use the I.465/V.120 protocol defined by the CCITT for such connections.

LAPD provides unacknowledged and acknowledged information-transfer services. The frame structure is essentially that of X.25: bit-oriented frames that start and end with an 8-bit flag that is avoided inside the frame by bit-stuffing; a 16-bit CRC is used for error detection; 16-bit addresses are used to

distinguish users connected to the same interface and different connections with a given user (i.e., different service access points). The unacknowledged service is implemented as a datagram; erroneous frames are discarded. The acknowledged service is implemented as a virtual circuit with Go Back N link error control. The receiver can turn the sender off and on by sending it “receiver not ready” and “receiver ready” frames.

The I.465/V.120 data link protocol is a modified version of LAPD that provides asynchronous data transfer, HDLC synchronous data transfer, and bit-transparent asynchronous transfer. To use this transfer protocol, the users first set up a circuit on the B channel by using the D channel. When the transfer is complete, the users release the circuit also by using the D channel.

The network layer of ISDN specifies the routing, multiplexing, and congestion control, in addition to the call-control messages.

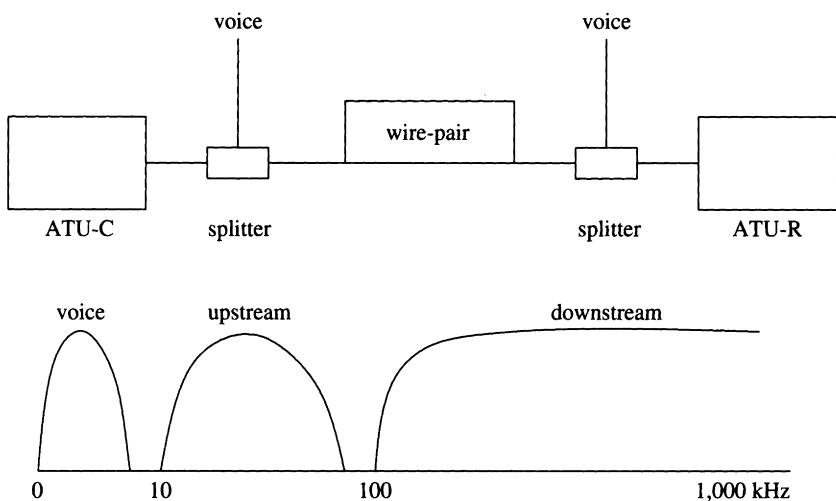
In summary, ISDN is an attempt to diversify the bearer services offered by the telephone network. The different services are provided by different networks (rather than in one single network) accessed through a common ISDN switch. Diversity is limited because these services are built on top of the traditional 64-Kbps channels, which constrains the bit stream rates that can be supported.

## 5.5.2 ADSL

Asymmetric Digital Subscriber Line (ADSL) is a modem technology that uses existing twisted pairs to create three channels: a high-speed downstream channel, a medium-speed upstream or duplex channel, depending on the implementation of the ADSL architecture, and a POTS (Plain Old Telephone Service) or an ISDN channel. The high-speed channel ranges from 1.5 to 6.1 Mbps, while duplex rates range from 16 to 640 Kbps. ADSL service providers offer different combinations of upstream/downstream bit rates, and many do not offer POTS. The channels may be further subdivided using TDM into 64 Kbps or ISDN services.

We first describe the physical layer and then the additional requirements for the network layer.

There is an ADSL modem at each end of the wire-pair, one at the subscriber end and the other at the central office. The central office modem is called ATU-C (ADSL Terminal Unit-Central Office); the subscriber modem is called ATU-R (Remote). The 1-MHz bandwidth of the wire-pair is divided into three regions. Frequencies below 4 kHz are split off into a separate channel for voice (if POTS is provided). Frequencies above 4 kHz are divided into an upstream or duplex channel (10–100 kHz) and a higher frequency downstream channel



5.23

FIGURE

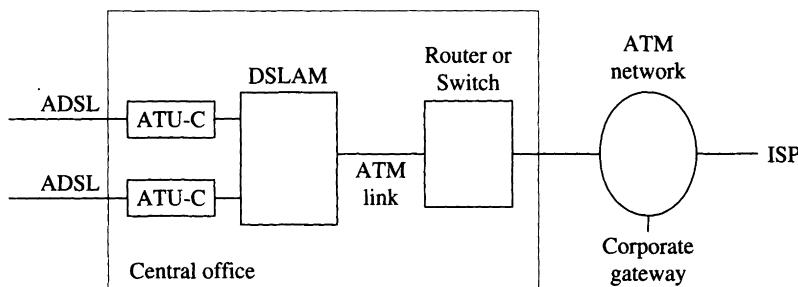
ADSL modems divide the bandwidth of the wire-pair into three regions.

(100–1,000 kHz). Thus ADSL is a passband technique, bypassing the 0 to 4 kHz voice band. By contrast, ISDN uses baseband modulation, overlapping with the voice band. (See Figure 5.23.)

A modulation scheme converts the serial bit stream into an electrical signal suitable for transmission. The ANSI ADSL standard specifies Discrete Multitone (DMT) modulation in which the 1-MHz bandwidth is partitioned into 256 4-kHz subchannels. A separate carrier in each band is quadrature-amplitude-modulated. A competitive, less complex technique called *carrierless amplitude-modulation-phase modulation* or CAP has recently emerged. In this technique there is a single downstream and a single upstream channel.

Channel performance over a twisted pair can vary greatly depending upon the length and condition of the wire. A pair is unsuitable for ADSL if it includes a loading coil, or if it terminates in a 4-kHz band-limiting digital loop carrier (DLC). Perhaps one-third of all pairs are unsuitable for this reason. The performance of the remaining pairs is highly variable depending again upon their length and the extent of far- and near-end cross-talk. Overcoming these impairments requires more sophisticated signal processing techniques or accepting lower bit rates.

To transport data packets over ADSL requires link layer protocols. There are two ANSI standards. The first uses Point-to-Point Protocol (PPP) variable-length data units within an HDLC framed structure (RFC 1662). The second



5.24

FIGURE

ADSL subscribers are connected via the DSLAM to a backbone network.

follows the ATM Forum's standard for ATM Frame UNI, also within an HDLC framed structure.

An ADSL service provider locates a Digital Subscriber Loop Access Multiplexer (DSLAM) in the central office. (If POTS is also provided, the voice signal split off at the ATU-C is sent to the telephone switch.) The DSLAM concentrates a number of ADSL subscriber lines to a single ATM line connected via a router or layer 2 switch to the provider's ATM backbone network through which a subscriber is connected to a corporate gateway (telecommuting applications) or an Internet service provider (Figure 5.24).

As with cable TV, ADSL service needs an expensive modem. A technician is needed to test if the existing wire-pair is suitable for ADSL. A new proposal called *DSL-lite* suggests a splitter-less ADSL. The objective is to define a service that makes installation easy, without requiring a technician. DSL-lite will not offer the full ADSL speeds, but will operate at speeds of up to 1.5 Mbps downstream.

There are several other variants of DSL. An earlier technology, high data rate DSL (HDSL), requires two twisted pairs, capable of 1.5 Mbps in each direction. (HDSL uses baseband signaling.) Rate-adaptive DSL (RADSL) modems test the line at start-up and adapt their operating speed to the fastest the line can handle. Very high data rate DSL (VDSL) modems operate at data rates up to 50 Mbps over short twisted pair connections.

Although the basic DSL technology was developed at Bellcore 15 years ago, it was ignored by the telephone industry until the emergence of two competitive forces. Competitive local exchange carriers or CLECs came into existence with the passage of the 1996 Communications Act and began to offer DSL service. More important was the threat posed by cable TV companies that began to offer Internet access. In reaction, the incumbent telephone companies

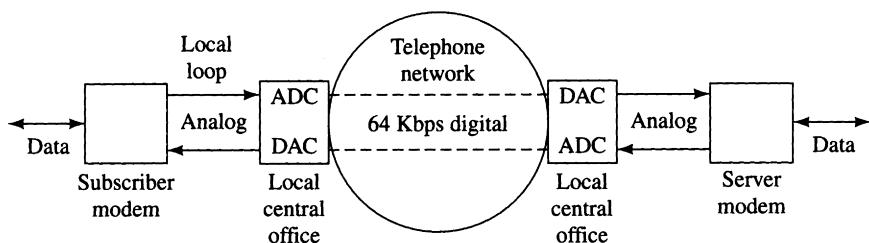
are now selling ADSL, priced aggressively to undercut both cable TV and CLEC offerings. The ADSL market segmentation that telephone companies are attempting is discussed in Chapter 10.

### V.34 and V.90 Modems

Most subscribers gain Internet access from their homes via analog voice modems that transmit their data over the analog local loop. The bandwidth of this loop is limited to 3 kHz. At the subscriber end, the modem converts binary data into an analog signal. At the local central office, the analog signal is sampled and encoded into a 64 Kbps digital signal. This analog-to-digital conversion (ADC) introduces quantization noise and limits the binary data rate to about 30 Kbps.

The 64 Kbps digital signal generated at the local central office is sent through the telephone network, again converted into an analog signal, transmitted over another local loop, received by the server modem, where the original binary data stream is recovered. The arrangement is shown in Figure 5.25. The ITU standard for this modem is called V.34.

The symmetric arrangement of the figure implies that downstream traffic (from server to subscriber) is also limited by the ADC conversion at the server's central office, so that downstream traffic is also limited to 30 Kbps. ISP servers today, however, bypass this ADC conversion, and their modems generate digital signals at 64 Kbps that reach the subscriber's central office. The digital-to-analog (DAC) conversion that occurs there is lossless, and so subscribers can receive 64 Kbps in the downstream direction (although upstream traffic is still limited to 30 Kbps). In actual fact nonlinearities and noise introduced in the DAC at the subscriber's local office limits the downstream speed to 56 Kbps.



**FIGURE**  
5.25

The arrangement for the V.34 Kbps modem. The ADC conversion limits speed to 30 Kbps. In the V.90 modem the ADC conversion at the server end is avoided, permitting a downstream speed of 56 Kbps.

The new standard is known as V.90. Among the interesting features of this standard are protocols that determine whether the subscriber modem and the local loop are able to support the Kbps rate. If they cannot, the server modem reverts to a V.34 modem.

## 5.6 INTELLIGENT NETWORKS

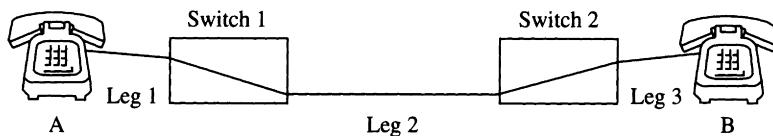
In our historical summary in section 1.1.1 we noted that modern switches in the telephone networks are programmable computers, which makes them very flexible. By sending instructions to a switch, one can modify its configuration. This contrasts with the earlier electromechanical switches, whose functions were built into their hardware. In modern switches, the control is separated from the hardware that executes the elementary switching operations.

This separation of control and basic operations is also present in the other network elements. As a result, these elements, too, are programmable. The separation enables telephone companies to develop their own services and implement them on switches and other network elements obtained from different equipment vendors.

*Intelligent Network* or IN is the name given to this network of programmable elements, organized to facilitate introduction of new services. In this section we explain an architecture model for intelligent networks proposed by telecommunication engineers. Since 1985, the IN concept has evolved through several proposed architectures. A number of telephone companies have implemented their own version of IN. Some wireless network operators are also implementing IN for mobile subscribers. Finally, the capability to implement new services that IN offers to a telephone company can in part be delegated to customers who can use the capability to design their own services. In section 5.6.1 we examine services that the telephone companies provide. We explain how the provision of a service can be decomposed into a sequence of basic steps. That is, service provision can be viewed as a script, or program, whose elementary steps are operations performed by the network elements. In section 5.6.2 we examine an architecture model for intelligent networks. In section 5.6.3 we list the basic actions the network performs when it implements a service.

### 5.6.1 Service Examples

Figure 5.26 illustrates the plain old telephone service or POTS. POTS is the basic telephone call service. When the network implements a telephone call, its elements execute the sequence of steps listed in the figure. In the old



- Steps in POTS:
1. Determine route from dialed digits
  2. Create and join legs 1, 2, 3
  3. Verify called party is available
  4. Conversation between A and B
  5. Detect termination by participant set "on-hook"
  6. Free legs 1, 2, 3

### 5.26

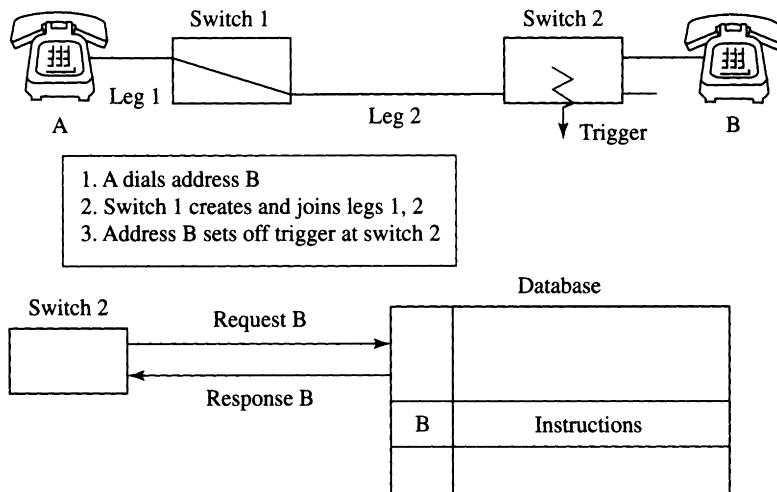
#### FIGURE

The figure notes the basic steps necessary to provide POTS or plain old telephone service.

telephone network, with electromechanical switches, the call functions of the network elements are wired into those elements. For example, when a user dials a telephone number, the successive digits generate a string of impulses that activate a sequence of rotary switches in the central offices of the telephone company. When an electrical current is sent to an on-hook phone, this current activates the bell. This arrangement of the hardware is inflexible. For instance, a customer cannot instruct the switches to block calls dialed from specific numbers or to forward a call to another number at certain hours of the day.

Figures 5.27 and 5.28 illustrate the operations performed by the modern network when it implements the *call forwarding* service. A customer using call forwarding can instruct the telephone network to forward an incoming call to another telephone set when the normal phone is not picked up after it rings (say) three times.

Figure 5.27 shows two telephone sets A and B connected via two switches 1 and 2. The customer at set A calls set B. The customer at B has instructed the telephone network to forward a call to some set B' after three rings. This instruction is stored in a database of switch 2 to which set B is attached. Moreover, the phone number of set B is stored in a *trigger table*. When a call for a telephone set reaches switch 2, the switch checks its trigger table. If that set is not in the trigger table, then the call is handled in the standard way. If the set is in the trigger table, then the switch places a request to its database. The database responds to the request by providing the instructions that the switch must follow to handle the call. Thus, when the call for set B reaches switch 2, the switch is triggered, and it requests the instructions from its database. Note that the telephone network can implement such services because the controls are separated from the actual operations of the switches.



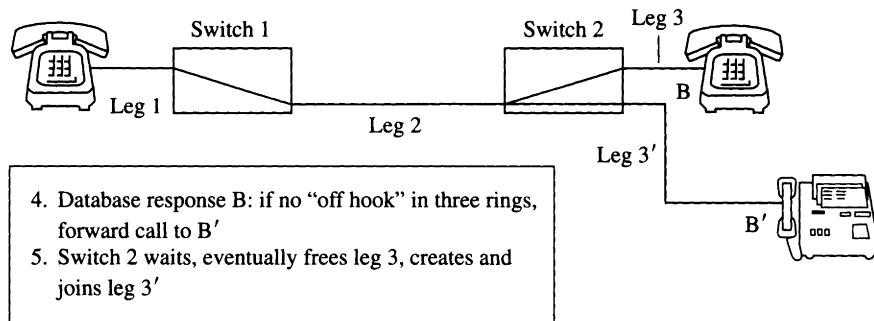
5.27

FIGURE

Call forwarding: when a call for B reaches switch 2, the switch searches a database for instructions.

### 5.6.2 Intelligent Network Architecture

The *Intelligent Network Architecture* (INA) is a model for organizing the programmable network elements and the communication between those elements. The objectives of that model are in part the same as those of the OSI reference model



5.28

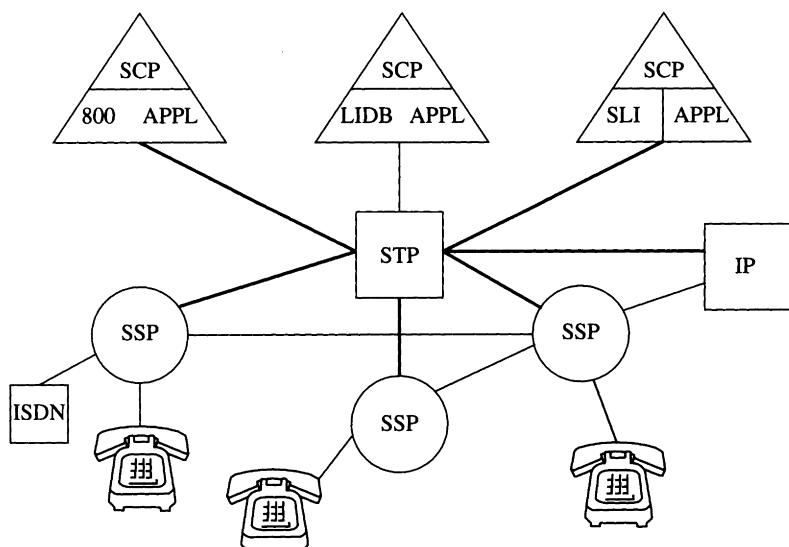
FIGURE

Call forwarding, continued: if the set B is not picked up after three rings, the call is forwarded to B'.

for computer networks: decomposition and standardization. INA is more complex in that it involves not only communication protocols and data elements as OSI does, but other network resources, including hardware elements. On the other hand, the OSI model is quite abstract, whereas the INA architecture is specifically designed for telephone networks.

As we discussed above, the main feature of an intelligent network is that the *control functions* and the *network resources* are separated. The network has transmission and other resources, including subscriber lines, trunk lines, switch ports, databases, and voice recorders. The control functions are call-control functions and resource-control functions. Examples of these functions include connecting three network users in a conference call, playing a recording, and collecting digits dialed by a user. To implement a control function, the network executes a sequence of atomic operations called *functional components*.

INA classifies the network elements into three types: *service switching points* (SSPs), *service control points* (SCPs), and *intelligent peripherals* (IPs). (See Figure 5.29.) These elements exchange information over the *common channel signaling* (CCS) packet-switched network, using the *signaling system 7* (SS7) protocol. INA calls the CCS network switches *signal transfer points* (STPs).



5.29

FIGURE

INA is a model for organizing network elements and the communication between them.

The figure shows three telephone sets and one IP attached to the network of SSPs. Regular lines denote the links used by the voice signals. The bold lines represent links of the CCS network. The SCPs contain databases and instructions for special applications.

The network elements are capable of performing different functions. The SSP can detect the need for IN service based on an originating line trigger such as off-hook, triggers applying to all calls such as for 800 or 911 calls, and terminating line triggers such as for call forwarding users. When such a trigger is activated, the SSP sends a request to the SCP for instructions. The SSP can identify, monitor, and allocate transmission resources (legs) connected to it. An IP can identify, monitor, and allocate nontransmission resources connected to it. The SCP detects IN service requests forwarded by the SSP. It then interprets the request according to a *service logic program* (SLP). A *service logic interpreter* (SLI) executes the SLP. A given SLI can execute multiple SLPs concurrently. The execution of the SLP involves invoking functional components and monitoring resources. Finally, at completion of the execution, the SLI notifies the SSP.

### 5.6.3 Functional Components

We briefly describe the functional components, the atomic actions on network resources, in five types of network operations.

#### *Control of Processing*

The functional components for control of processing are of two types: to provide instructions when the SSP asks the SCP to take control of the call processing and to effect the release of control when the SCP returns the control to the SSP after servicing the request.

#### *Connection Request*

A connection request involves the following functional components: creating a leg between an SSP and another network element, joining a leg to an ongoing call, splitting a leg from an ongoing call, and freeing a leg to release the resource.

#### *User Interaction Request*

Two types of functional components are invoked in user interactions: sending information such as a prerecorded announcement to a call participant and receiving specific information such as dialed digits from a call participant.

### ***Network Resource Status Request***

Network resource status requests are used by the SLP in processing some call control. Monitoring is a functional component that instructs the SSP for notification of a particular event on a specified leg, such as on-hook, flash-hook, and off-hook.

### ***Network Information Revision Request***

Network information revision requests enable the SLP to change the data stored in the SSP tables.

In summary, INA is a culmination of a long development in which network element functions or operations are separated from the control of those functions. This separation permits the creation of new services as programmable sequences of functional components. Sophisticated customers can program these sequences by themselves. A very important example is 800 number services. Companies, such as credit card and direct order companies that provide direct customer services over the 800 number phone, can route customer calls to different parts of the country or to different operators depending on the time of day, the subscriber's location, and other information provided by the subscriber via the telephone keypad.

---

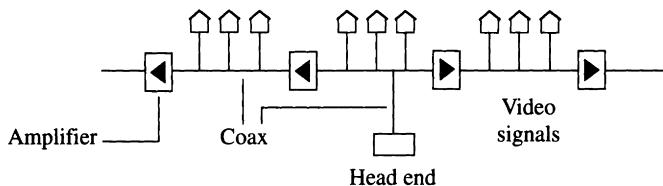
## **5.7**

## **CATV**

CATV, originally *Community Antenna TV*, now refers to any cable or hybrid optical fiber and cable (HFC) system used to deliver video signals to a community.

The standard CATV distribution system is illustrated in Figure 5.30. The head station distributes analog video programs. Several programs are frequency-division multiplexed at the head end over a coaxial cable network. Each program occupies a 6-MHz channel, with the spectrum between 50 and 550 MHz accommodating up to 80 channels. Every subscriber receives the same programs.

Three innovations have transformed CATV from a video distribution system to one that can provide interactive, integrated services. The first innovation, discussed in section 5.7.1, upgrades CATV into a two-way communication system. The second introduces link layer functions that provide users access to a shared data link. It is described in 5.7.2. The third consists of digital compression schemes that enable video to be transmitted at relatively low speeds. In section 5.7.3 we discuss three services that CATV operators may offer: Internet



5.30

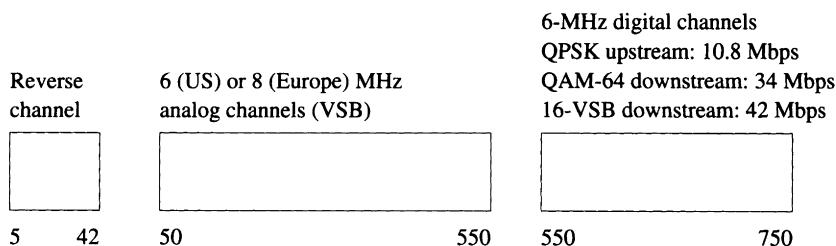
**FIGURE**

In the standard CATV network, the head station distributes the video programs over a coaxial cable network. Amplifiers are inserted into the distribution cable in order to boost signal power. All subscribers receive the same programs.

access, video on demand, and telephony. In section 5.7.4 we elaborate on the MPEG standards used for video compression.

### 5.7.1 Layout

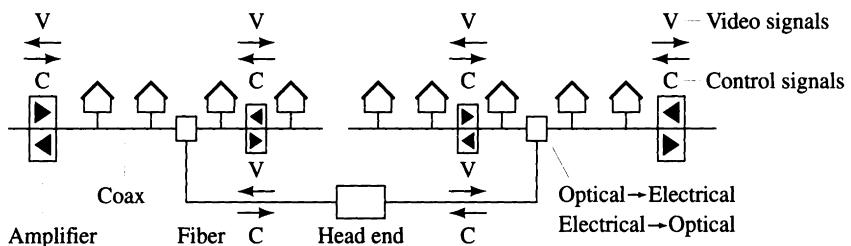
The physical layout of an upgraded network is sketched in Figure 5.31. The network between the head stations and the users consists of optical fiber terminating at a fiber node to which is attached a local coaxial network that connects to 500 homes. The head stations of the service providers have access to video servers, web servers, and the Internet, via a backbone network that is not shown. The total bandwidth is increased to 750 MHz, as shown in Figure 5.32. The more important difference, seen in Figure 5.31, is that signals can travel in both directions. The fiber node converts the downstream optical signal



5.31

**FIGURE**

The network connects the head end stations of the service providers to the user equipment. The downstream signal is carried over an optical fiber to the fiber node, where it is converted into an electrical signal and distributed over coaxial cable. User upstream signals use the same physical network.

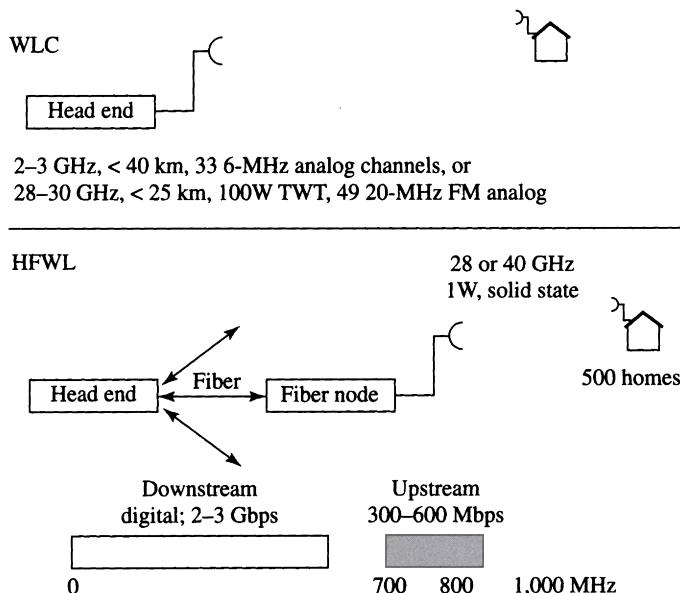


**FIGURE**  
5.32

The network connects the head stations of the service providers to the user equipment. Signals are sent downstream from head stations to users, and upstream to head stations.

originating at the head end into an electrical signal, and the upstream electrical signal originating at the users into an optical signal.

Figure 5.33 gives an alternative distribution technology in which all or part of the distribution system is wireless. In the wireless cable system, subscribers directly access the signal broadcast from the head end station. In the



**FIGURE**  
5.33

The wireless cable (WLC) system replaces the current distribution system. The hybrid/fiber wireless (HFWL) system replaces the local coaxial distribution.

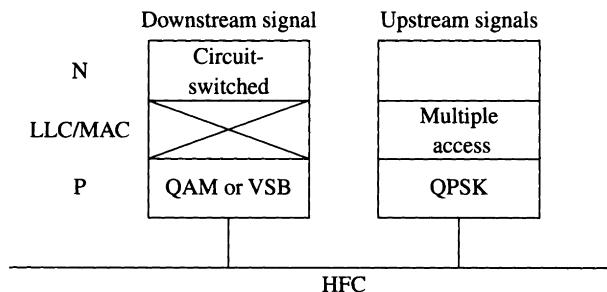
hybrid/fiber wireless system, a digital video signal is sent to the curb over optical fiber (as in Figure 5.31), and the local coaxial distribution system is replaced by a local wireless system. The wireless portion of these networks would extend over short distances. These systems may be less expensive than cable when there is a high geographical concentration of users.

### 5.7.2 CATV Layered Network

Figure 5.34 sketches a plausible three-layer decomposition of the functions of the CATV network. The decomposition is different for downstream and upstream traffic.

We begin with the physical layer. Figure 5.32 shows proposed uses of the available frequency spectrum of 5 to 750 MHz. The downstream and upstream signals occupy different frequency bands. This separation of frequency bands prevents the amplifiers from amplifying their own output, which would lead to saturation of the amplifiers and to oscillation.

The downstream signals occupy the spectrum from 50 to 750 MHz. Conventional analog broadcasts that can be received by existing television sets occupy 6-MHz channels between 50 and 550 MHz. The spectrum between 550 and 750 MHz may carry digital MPEG-2 programs, data streams, and downstream telephony. Using QAM-64 or 16-VSB modulators, each 6-MHz analog channel is converted into a digital link with a bit rate of 27 to 38 Mbps. Such a link can be used to carry 6 to 10 MPEG-2 programs at rates of 3.5 Mbps, or to transport digital data to users. The MPEG programs may be decoded by set-top boxes. Transmission of user data requires cable modems. The downstream



5.34

FIGURE

Plausible three-layer decomposition of network functions. The downstream signals are carried by a circuit-switched network. The upstream signals are sent using a multiple access scheme.

signals are all broadcast. As shown in Figure 5.34, the downstream signals are circuit-switched.

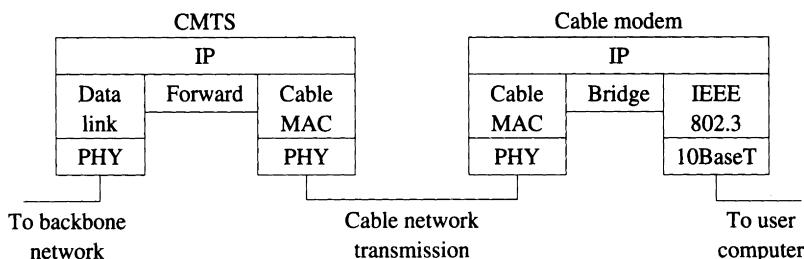
Upstream signals from users occupy the 5 to 42 MHz spectrum. This spectrum is usually divided into 2-MHz channels. Because the network has a tree and branch structure, the transmission path from users to the head end is shared. The effect is that the signal received at the head end is the sum of the user signals, so a MAC protocol is needed to provide collision-free access, as is described in the next paragraph. Moreover, the tree and path structure causes the addition of noise from all the users. Both narrow-band noise from electromagnetic radiation and nonlinearities, and impulse noise from appliances, are present. The bits are modulated using quadrature phase-shift keying (QPSK) and forward error-correction is used, yielding a shared link with a bit rate of about 3 Mbps for each 2-MHz channel. The advantage of QPSK is that by choosing the carrier frequency of the sine waves judiciously, the network engineer can make sure that the modulated bit stream does not interfere with the video signals. Another advantage is that this modulation scheme requires only a small range of frequencies to transmit the bits. Only a small range of frequencies is generated because the modulated signal changes only once every  $n$  bits.

For transfer of data between users and the head end, cable operators (called *Multiple Service Operators* or MSOs) offer a shared 3-Mbps (over a 2-MHz channel) upstream link and a broadcast 38-Mbps (over a 6-MHz channel) downstream link. There are several proprietary MAC protocols for access to the shared link. An emerging standard, called *Data-Over-Cable Service Interface Specifications* (DOCSIS) is being developed by the Multimedia Cable Network Systems (MCNS) consortium. The goal is to transparently transmit IP traffic between the user *cable modem* and the *cable modem termination system* or CMTS at the head end. We briefly describe DOCSIS.

Upstream frames and downstream frames are synchronized (as in the PON systems), and cable modems use ranging measurements so that signals from different users do not overlap. The frames are divided into minislots, some of which are kept aside for user reservation requests. The request includes the modem ID and the amount of bandwidth requested.

When a user makes a request, the CMTS may grant the user a certain number of minislots in the next frame. The grants are carried in a downstream frame. However, more than one cable modem may request at the same time, resulting in a collision. The colliding modems learn about this, because they do not receive any grants. They must back off for a random amount of time before making another request.

The standard specifies how user Ethernet packets or ATM cells are to be framed.



5.35

FIGURE

The CMTS can forward user layer 2 or layer 3 packets to the backbone network.

The cable modem is connected via the CMTS to a backbone network. The CMTS may serve as a layer 2 bridge or provide network layer connectivity, as shown in Figure 5.35.

### 5.7.3 Services over CATV

The large bandwidth in the downstream broadcast direction can support not only 80 traditional, 6-MHz TV channels, but also several hundred MPEG-2 channels. This possibility is realized as a service called *video on demand* or *video dial-tone*. Subscribers browse through a large collection of video programs and request a program, using their set-top boxes. The head end transports the requested MPEG digital stream over an available channel. The set-top box demodulates and decompresses the received bit stream and generates the NTSC or HDTV signal for display on the TV set.

The initial enthusiasm for this service was greatly reduced following an unsuccessful field trial by Time Warner. The lack of success may have been due to the high cost of the set-top boxes, or the near-substitutes (such as video rentals) that are available. Video on demand survives in an attenuated form in hotels and as pay-per-view.

A later development has been the use of CATV for Internet access. Internet service providers, together with cable operators, offer subscribers Internet access over a shared 3-Mbps upstream link and a 38-Mbps downstream broadcast link. Typically, 10 subscribers share the resources at any time. Providers enhance performance by Web caching and offer directory and other services. Subscribers must purchase a cable modem and usually pay a monthly flat rate. The growth of subscribers to this service has been impressive. However, because the links are shared, a few heavy users can cause congestion and degrade

service for all. This seems to be a common occurrence, and service providers are imposing restrictions on usage.

AT&T's acquisition of TCI is fueling speculation that subscribers will soon be offered telephone service over CATV using the next generation cable modems. The attraction for AT&T might be its ability to offer lucrative long-distance telephone service without paying access charges to the local telephone company.

Because the upstream channel is so noisy, a technician is needed to install a cable modem. This makes deployment of cable modems expensive. Wide deployment of cable modems will require technical advances that make them as easy to install as telephone modems.

### 5.7.4

### MPEG

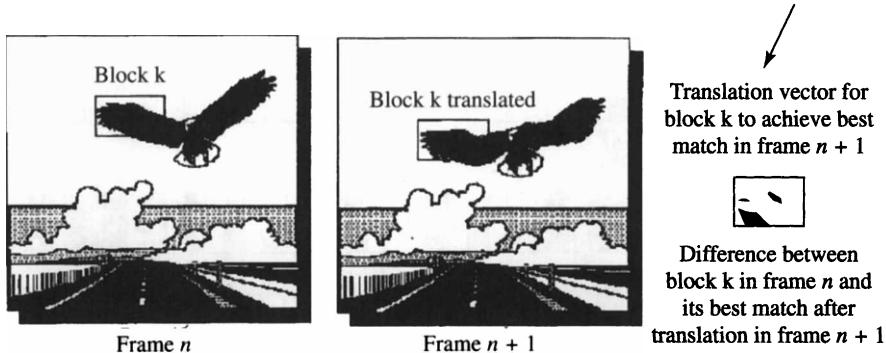
The Moving Pictures Expert Group has defined a number of standards for video compression. The MPEG compression algorithms use the redundancy within each frame and the similarity of successive frames. We give a brief historical account of the algorithms, starting with JPEG.

JPEG (Joint Photographic Experts Group) is the ITU protocol for coding continuous-tone still images. JPEG is used to show images that do not move, as in home shopping TV ads. JPEG employs a spatial digital cosine transform or DCT compression scheme.

H.261 is the ITU standard for videoconferencing applications at ISDN transmission rates. The coding scheme uses DCT, DPCM (Differential Pulse Code Modulation), and motion compensation. The motion compensation algorithm assumes relatively slow and restricted movement, so quicker movement appears jerky. The first frame is an I (intra-frame) frame; each subsequent frame uses inter-frame prediction (P frames), using only the nearest frame for prediction. The quantization step-size is adjusted to achieve a target bit rate.

The Motion Picture Experts Group (MPEG) was founded under ISO. MPEG-1 defines a coding scheme using DTC and motion compensation similar to H.261. It is targeted for use by progressively scanned media, such as CD-ROM. MPEG-1 provides frame-based random access of video, fast forward/fast reverse searches, reverse playback of video, and ability to edit. In addition to the I and P frames, MPEG-1 introduced B-frames, which use motion picture prediction based on the two nearest already coded frames.

Progressive scanning means the first line is displayed first, then the second, and so on. MPEG-2 is a superset of MPEG-1 with special consideration of interlaced scanning used in broadcast TV, and scalable video extensions for



5.36  
FIGURE

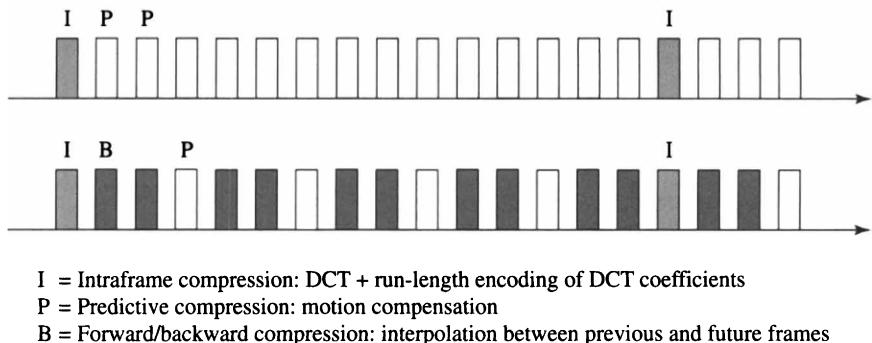
The motion-compensation algorithm compares successive frames. A frame is divided into blocks, and the corresponding blocks in the next frame are moved until they match those in the previous frame best. The algorithm transmits the block displacement vectors and the residual differences between the blocks.

digital TV and HDTV. Scalability is achieved by multiresolution representation that permits downscaling an input video signal into a lower resolution video.

Motion compensation in MPEG is illustrated in Figure 5.36. The figure shows two successive frames of a video. The algorithm decomposes frame  $n$  into blocks. The algorithm then examines frame  $n + 1$  and finds the translation of the blocks of that frame that makes them most similar to the corresponding blocks of the previous frame. Instead of transmitting all the blocks, the algorithm transmits the displacement vectors and the differences between a block in a frame and the corresponding block in the next frame after translation. The frames that contain this motion-compensation information are called *P frames*. Finally, an MPEG algorithm can use interpolation frames, called *B frames*. The *B frames* perform an interpolation between *P frames*.

Figure 5.37 shows the sequence of frames produced by MPEG-2. Refresh frames are sent twice every second to avoid error propagation. The number of *B* and *P* frames can be selected to produce different qualities and resulting bit rates.

MPEG-4 and its predecessor H.263/L assist low bit rate (64 Kbps) transmission with multimedia data access tools to facilitate indexing, downloading, and querying; to efficiently combine synthetic scenes; and to handle multiple concurrent data streams. H.263 is used in video streaming.



**5.37**  
**FIGURE**

Sequence produced by an MPEG-2 algorithm. This sequence consists of refresh frames (I), motion-compensation frames (P), and interpolation frames (B).

## 5.8

## SUMMARY

The bearer service offered by circuit-switched networks is fixed bit rate end-to-end connections with a delay equal to the propagation delay. This bearer service is well suited for constant bit rate traffic. The service can of course be used to support both variable bit rate and message traffic. However, used in that way the utilization of the network link capacities may be very low.

The telephone and CATV networks are the most important networks in terms of number of customers and ability to finance new investment. Recent innovations will enable these networks to provide new services that can propel them well beyond their traditional markets.

Innovations in the telephone network have led to (1) higher bandwidth links in the backbone network through SONET and WDM; (2) higher speed local loop using optical fiber or DSL over twisted pairs; (3) integrated services, providing a common interface for message transmission over packet-switched networks and constant bit rate circuit-switched connections; and (4) Intelligent Network Architecture, permitting a very flexible means to create new user services.

SONET implements the bit way layer of the Open Data Network model, taking advantage of the economies of scale offered by optical communication. The optical local loop, perhaps more expensive than today's copper wire, will bring to the user the high-speed access necessary for video applications. Lastly, DSL and INA exploit the economies of scope by integrating packet-

and circuit-switched services and other communication-related services such as call forwarding.

Innovations in CATV have made possible (1) a more efficient use of bandwidth by digital compression; (2) increased bandwidth by use of optical fiber for transmission to the curb; and (3) transmission of information by the subscriber. With these innovations CATV can begin to compete for Internet access and local telephone services.

Despite their innovations, packet-switched networks are best suited for message traffic, and circuit-switched networks are best suited for constant bit rate traffic. Both types of network are still not well suited for variable bit rate traffic. That traffic is best carried by ATM networks, which we study in the next chapter.

## 5.9

## NOTES

The SONET/SDH framing conventions are described in [BC89, S92, B95]. A discussion of SONET rings and other architectures that improve network survivability appears in [WL92, W95]. ATM over SONET is specified by the ATM Forum [A93]. PPP over SONET is discussed in RFC 1619. PPP provides a standard method for transporting multiprotocol datagrams over point-to-point links (RFC 1661). IP/PPP/SONET is discussed in [MADD98].

DWDM is discussed in [CM98a, JSAC98].

The AT&T SLC 5 is briefly described in [C89]. The British TPON system is described in [R91]. APON is discussed in [VVM93, ALFV98]. The passive photonic loop is described in [WL89]. Promising new local loop technologies that can provide high-speed access are described in [CM91]. An informative comparison of the costs of different broadband access technologies (CATV, ADSL, hybrid systems) is given in [PB95].

ISDN standards are described in [S92]. See [F95, P95] for a recent appraisal of the market penetration of ISDN.

ANSI standards for xDSL are available at the Web site of the ADSL Forum, [www.adsl.com](http://www.adsl.com). A historical overview of xDSL is provided in [H97]. The modulation technologies for xDSL are discussed in [JSAC95].

An issue of *Communications Magazine* [CM92] contains good discussions of IN in the United States and several other countries.

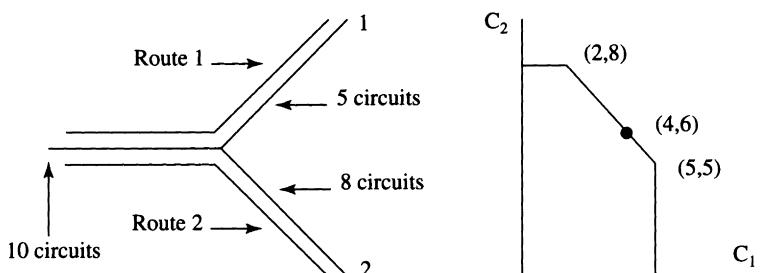
The CATV industry is developing standards called Data Over Cable Service Interface Specification (DOCSIS). MPEG standards are discussed in [S96]. Physical layer and MAC protocols are also being developed by the IEEE 802.14

working group. For a description of video dial tone, see [CM94] and [IN95]. ADSL and cable modems are seen as competitive technologies, and both industry groups make very optimistic projections.

## 5.10

## PROBLEMS

1. For a network like the one in Figure 5.1, let  $\{C_{ij}\}$  denote link capacities. Suppose  $C_R = \{C_r\}$  is a set of route calls, and say that  $C_R$  is feasible if (5.2) holds. If there are  $N$  routes in  $R$ , then  $C_R$  is an  $N$ -dimensional vector.
  - (a) Show that the set of feasible vectors is convex.
  - (b) Suppose that a connection over route  $r$  brings a revenue of  $p_r$  per unit of time. Show that the set of route calls  $C_R^*$ , which maximizes revenue, can be obtained as a solution of a linear programming problem.
  - (c) As each call request arrives, the call admission algorithm decides whether to admit or reject the call. We would like to design an algorithm that maximizes revenue. This problem illustrates the difficulties in designing such an algorithm. Consider the network shown in Figure 5.38 with three links, with capacities as shown, and with two routes. Show that the set of feasible calls is given by the convex region on the right. Design an admission procedure so that the system will operate near the desired point  $(4,6)$ . Consider two cases. In the first case, existing calls can be terminated (so-called preemptive case); in the second case, existing calls cannot be terminated (nonpreemptive case).
  - (d) Planning problems are concerned with expanding the network capacity to meet growing demand. Suppose that this growth is described as



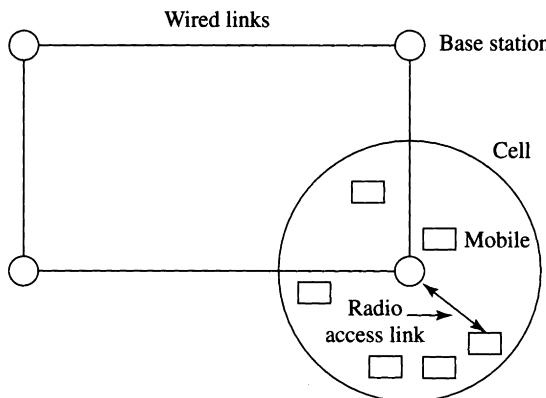
5.38

FIGURE

The panel on the left shows a network with three links and two routes. The set of feasible calls is the convex region shown on the right.

an increase of  $\Delta_r$  calls along route  $r$ ,  $r \in R$ . Suppose this increase is to be met by expanding the capacity of each link  $(i, j)$  by amount  $\delta_{ij} \geq 0$  at a unit cost of  $p_{ij}$ . Formulate the minimum cost expansion as a linear programming problem.

2. In the preceding question, it is assumed that the set of routes has already been selected and the number of simultaneous calls  $C_r$  on each route  $r \in R$  is given. But the route selection and assignment may be changed. Suppose the link capacities  $C_{ij}$  are given. Suppose we wish to route  $N_{xy}$  simultaneous calls from every originating switch  $x$  to destination switch  $y$ . Formulate a linear programming problem whose solution gives a routing assignment.
3. A cellular telephone network consists of a number of base stations (switches) connected by wired links. A user within a cell served by a particular base station gains access to that station using an idle radio or wireless link. If all the access links are busy, the user's request is blocked. (See Figure 5.39.) The number of access links to each base station is fixed by government regulations. The number of mobile users varies randomly.
  - (a) Formulate a model to determine the blocking probability.
  - (b) Suppose a call is placed from a mobile in one cell to another mobile in another cell. The route for such a call would have three parts: a radio access link in the first cell, a route over the wired links from the first base station to the second, and a radio access link in the second cell.



5.39

FIGURE

The base stations or switches in a cellular network are connected by wired links. Mobile stations in a cell share wireless access links.

What is the blocking probability of the call in terms of the blocking probabilities for each part of the route?

4. The textbook time-division multiplex scheme assumes that all the multiplexed signals have identical bit rates or frequency (see Figure 2.5). Since each signal comes from a separate source, this assumes that the clocks at all those sources are perfectly synchronized. This is impossible in practice, and so, over time, the clocks will drift apart. The more accurate the clocks, the smaller will be the drift. If we assume that the clock accuracy is on the order of  $10^{-n}$  (i.e., they drift apart by 1 out of every  $10^n$  seconds), and if we assume a bit rate of  $10^m$  bps, then these signals will drift by 1 bit every  $10^{n-m}$  seconds. For example, if we assume a clock accuracy of  $10^{-12}$  (very high) and bit rate of  $10^9$  bps, there is a drift of 1 bit every  $10^3$  seconds (about 1 bit per hour). If we assume a clock accuracy of  $10^{-4}$  and bit rate of  $10^6$ , there is a drift of 100 bits every second. Use those concepts to analyze how frequently it will be necessary to use the frequency justification procedures of Figure 5.11.
5. The frame structure in X.25 uses an 8-bit flag. Suppose it is 01111110. This 8-bit pattern may be present in the data, in which case it may be interpreted as the flag. To avoid this confusion, whenever a pattern of six 1s appears in the data, it is followed by a seventh stuffed bit of 1. Show that the data will never carry the flag pattern. Also explain how the original data can be recovered.
6. Design a caller ID service using the functional components of section 5.6.3.

---

---

# 6

---

---

## CHAPTER

# Asynchronous Transfer Mode

In Chapters 3 through 5 we studied packet- and circuit-switched networks. Those networks are suitable for message and constant bit rate traffic, respectively. In this chapter we examine the Asynchronous Transfer Mode or ATM networks. ATM networks combine the good features of both types of networks, making them suitable for both constant bit rate (CBR) and variable bit rate (VBR) traffic. ATM networks potentially can provide bearer services with a specified quality of service to meet the needs of all traffic types. Whether this potential can be realized depends on how well the problems of management and control of ATM networks can be solved. Those problems are discussed in Chapters 8 and 9.

This chapter presents the concepts of ATM. It offers a simple model to calculate the delay of an ATM network. By the end of this chapter you will understand the ATM layered architecture, the addressing and routing standards, and the formats for different services. You will also know the important proposals for ATM LANs and for IP service over ATM. With these proposals, ATM becomes backward-compatible with existing LAN equipment and IP software.

We saw in section 2.2 that different applications impose different performance requirements on the network bearer services in terms of delay, bandwidth, and loss. If all these applications are to share the same network resources (links, buffers, switches)—and this is very desirable to gain the economies of scale and service integration—the network must be able to allocate its resources differently to different applications. Because switches or routers in datagram networks do not have connection state information, they cannot differentiate among packets by application. Therefore, datagram networks cannot

discriminate among applications by quality of service. Circuit-switched networks do maintain connection information. But since they provide a fixed set of resources to every connection or call, they, too, cannot differentiate among connections.

In the mid-1980s some engineers argued that virtual circuits were ideal for the efficient utilization of network resources when applications have widely different performance requirements. In a virtual circuit network, the nodes can set aside resources for specific connections, and they can also discriminate among connections in order to meet their different requirements. Those arguments culminated in the development of a new set of standards for a class of virtual circuit networks called ATM networks. ATM networks seek to provide the end-to-end transfer of fixed-size packets or cells over a virtual circuit and with specified quality of service (in terms of delay, speed, and error rate).

We describe the main features of ATM networks in section 6.1. In section 6.2 we discuss addressing, signaling, and routing in ATM networks. In section 6.3 we present the structure of the ATM header. In section 6.4 we discuss the ATM adaptation layer, which converts information streams in a variety of forms into a sequence of ATM cells. In order to satisfactorily serve a wide range of applications, ATM networks must be appropriately controlled. Standards relating to the management and control of ATM networks are discussed in section 6.5. ATM networks are designed to support both real-time applications such as video connections and telephone services and non-real-time applications such as e-mail and file transfers. In section 6.6 we study how ATM may support Broadband Integrated Services Digital Networks (BISDNs). In section 6.7 we discuss internetworking with ATM. Our objective is to explain how the familiar TCP/IP applications can be supported by an ATM network. Section 6.7 also covers LAN emulation by ATM networks.

We provide a summary in section 6.8.

## **6.1 MAIN FEATURES OF ATM**

Four features distinguish the ATM bearer service from other bearer services:

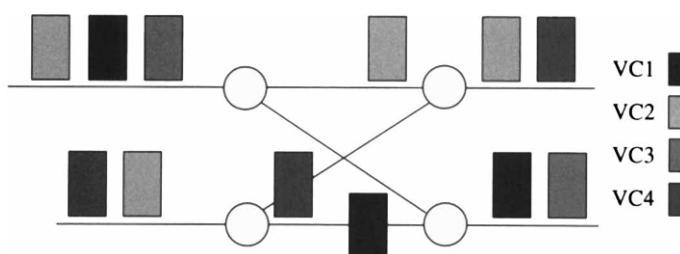
1. the service is connection-oriented, with data transferred over a virtual circuit (VC);
2. the data is transferred in 53-byte packets called *cells*;
3. cells from different VCs that occupy the same channel or link are statistically multiplexed;

4. ATM switches may treat the cell streams in different VC connections unequally over the same channel in order to provide different qualities of service (QoS).

We emphasize that ATM is a *bearer* service and *not* a bit way in terms of the Open Data Network model of section 2.8. The actual transfer of the bits that constitute the cell may be carried out in very different bit ways, ranging from SONET to DS-3 channels to proprietary bit ways. We now discuss the advantages and disadvantages of the four features listed above.

### 6.1.1 Connection-Oriented Service

In a connection-oriented service over a virtual circuit, the data stream from origin to destination follows the same path. (See Figure 6.1.) Data from different connections is distinguished by means of a virtual path identifier (VPI) and virtual channel identifier (VCI). A connection over a virtual circuit is called a *virtual channel* in the ATM terminology. Thus each cell incurs an overhead corresponding to the length (number of bits) of the VPI/VCI, which is generally much smaller than the length of a full source/destination address needed in a datagram service. (There is additional overhead as well. Together with the small payload, the overall overhead/payload ratio may be higher for ATM than, say, with IP.) Second, cells in the same connection reach the destination in the order they are sent from the source, thus eliminating the need for sequence numbers and for buffering packets at the destination if they arrive out of order. (However, sequence numbers are necessary if the application layer must detect cell loss. We see this in our discussion of the adaptation



6.1

FIGURE

In virtual circuit transport, the cells of a given connection follow the same path in the network.

layer, in section 6.4.) More importantly, ATM switches can identify different connections by their VPI/VCI. Consequently, the switches potentially can discriminate among different connections.

This potential can be used in many ways: admission control (refusing certain connections if sufficient network resources are unavailable), congestion control (limiting the amount of traffic accepted from a connection), resource allocation (negotiating the bandwidth and buffers allocated to a connection), and policing (monitoring the burstiness and average rate of traffic in a connection). We will study these different modes of control in Chapters 8 and 9.

The main disadvantage of connection-oriented service is that the network must incur the overhead of connection setup even when only a few cells are to be transferred, which could be done more efficiently by a datagram service. Another disadvantage is that a link or node failure terminates the virtual channel, whereas such a failure affects only a few packets in a datagram network. (However, the ATM Forum will be releasing specifications of edge-based routing that automatically reroutes a connection in case of failure, minimizing cell loss without requiring the user to reestablish the call.)

The ATM Forum currently specifies five categories of services that an ATM network can provide: constant bit rate (CBR), variable bit rate-real time (VBR-RT), variable bit rate-non-real time (VBR-NRT), available bit rate (ABR), and unspecified bit rate (UBR). A sixth service category, guaranteed frame rate (GFR), was recently proposed in the ATM Forum to ensure minimum rate guarantees for UBR services.

These services differ in the parameters of the traffic and of the quality of service that they specify. The parameters of the traffic are defined by an algorithm—called the *generalized cell rate algorithm* (GCRA)—that controls the arrival times of cells. These parameters are the following:

- ◆ peak cell rate (PCR),
- ◆ sustained cell rate (SCR),
- ◆ initial cell rate (ICR),
- ◆ cell delay variation tolerance (CDVT),
- ◆ burst tolerance (BT),
- ◆ minimum cell rate (MCR).

We examine these parameters in Chapter 8. For now, here is a brief discussion of their meaning. PCR is the reciprocal of the minimum time between two cells. SCR is the long-term average cell rate. ICR is the rate at which a source should send after an idle period. CDVT measures the permissible departure

from periodicity of the traffic. BT measures the maximum number of cells in a burst of back-to-back cells. MCR is the reciprocal of the maximum time between two cells.

For CBR traffic, PCR measures the maximum rate, and CDVT specifies the acceptable jitter, that is, the departure from strict periodicity at rate PCR. For VBR traffic, the key parameters are SCR and BT, which measure the long-term rate of the traffic and its burstiness. For ABR traffic, the cell rate is between MCR and PCR. For UBR, no parameters are controlled, but no QoS is guaranteed.

Thus, CBR is an essentially periodic stream of cells with some acceptable jitter that may be unavoidable because of framing or packetization. VBR is a bursty traffic that may be generated by a variable bit rate codec. ABR and UBR are for irregular data traffic. GFR is proposed as an enhancement to UBR service.

The quality of service (QoS) parameters (attributes) are the following:

- ◆ cell loss ratio (CLR),
- ◆ cell delay variation (CDV),
- ◆ peak-to-peak cell delay variation (peak-to-peak CDV)
- ◆ maximum cell transfer delay (Max CTD),
- ◆ mean cell transfer delay (Mean CTD).

These service parameters may be negotiated between end systems and the network when a connection of a particular service category is being set up. There also are "non-negotiated" QoS attributes including cell error ratio (CER). The service categories specify the traffic and QoS parameters according to Table 6.1.

We should think of the traffic parameters for a particular service as placing restrictions on the traffic generated by a user, and the QoS parameters for that service as obligations on the network that provides the service.

The UBR service is best effort and requires only the selection of one path from the source to the destination. To provide CBR, VBR, and ABR services, the ATM switches must reserve resources when the call is set up. In the case of ABR, as we discuss in Chapter 8, the flow of cells is regulated based on feedback information about the network congestion. CBR and VBR traffic are regulated at the source (by the GCRA) without any network feedback.

The proposed GFR service is not included in Table 6.1. GFR is intended for applications that can neither specify the SCR or BT needed for VBR service, nor be subject to the rate-based control rules of ABR. Current internetworking applications (based on TCP/IP) fall into this category. GFR provides a minimum

ATM Layer Service Characteristics						
Attribute	CBR	VBR(RT)	VBR(NRT)	ABR	UBR	Parameter
CLR	Specified	Specified	Specified	Specified	Unspecified	QoS
CDT, CDV	CDV and Max CTD	CDV and Max CTD	Mean CTD only	Unspecified	Unspecified	QoS
PCR, CDVT	Specified	Specified	Specified	Specified	Specified	Traffic
SCR, BT	n/a	Specified	Specified	n/a	n/a	Traffic
MCR	n/a	n/a	n/a	Specified	n/a	Traffic
Congestion control	No	No	No	Yes	No	

6.1

TABLE

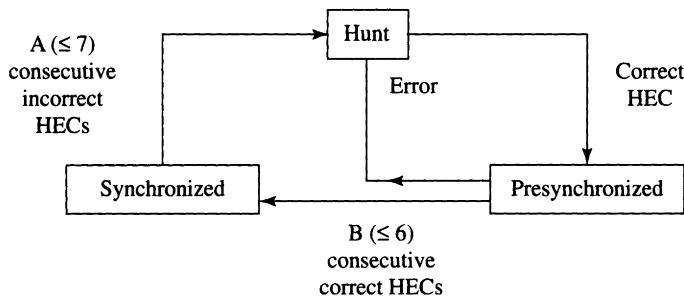
ATM service classes and applicable parameters. Source: [AL95, A96d].

frame (not cell) rate guarantee. The traffic parameters for GFR specify a maximum frame size for the CPCS-SDU packets (CPCS is discussed in section 6.4). If the user sends packets of smaller size at a rate smaller than the minimum guaranteed rate, they will be delivered by the network.

### 6.1.2 Fixed Cell Size

In order to recognize packet or cell boundaries in the bit stream transmitted by the physical link, it is customary to delimit packets by a distinguished bit pattern. However, if cells are of fixed length and if they contain a fixed-length error-checking sequence such as a CRC (which ATM cells do), then the node can use these features to determine cell boundaries implicitly. The basic idea is illustrated in Figure 6.2.

Suppose that each cell of length  $N$  contains a CRC sequence of length  $n$  computed over the preceding  $m = N - n$  bits. In ATM cells, this CRC field is the header error-control field (HEC). The HEC field is the last header byte, and it is calculated over the previous 4 bytes. The algorithm starts with any  $n$ -bit "window" as a tentative CRC and matches it with the CRC sequence computed over the preceding  $m$  bits. If a match occurs, the cell boundary has been correctly identified; otherwise, the algorithm shifts the window by 1 bit and repeats the procedure. A match occurs in at most  $N$  steps if the cell contains no errors. Once the cell boundary is identified, subsequent boundaries are found by counting bits. Conversely, if CRC errors are detected in several consecutive cells, this can be taken to indicate loss of synchronization (loss of cell boundary location).



## 6.2

**FIGURE**

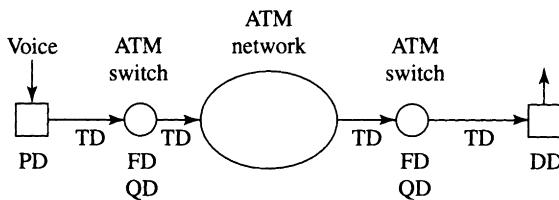
The figure summarizes the algorithm that a node uses to locate the cell boundaries. The node verifies the header error-control field (HEC) and searches for the cell boundary when it detects errors. Note the hysteresis of the algorithm designed to prevent it from hunting for a new boundary after every transmission error.

The relatively short size of the ATM cells implies a large overhead that takes up approximately  $5/53$  or 9.4% of the bandwidth of every link, which is a disadvantage. (Were the cell size doubled, this overhead would be under 5%) The short size of ATM cells was selected to reduce packetization delay for real-time voice data. Suppose voice is sampled 8,000 times per second, or once every  $125 \mu\text{s}$ , and suppose each sample is encoded into 1 byte. It then takes  $125 \times P \mu\text{s}$  to fill up a cell containing data of  $P$  bytes. Thus the packetization delay for ATM cells with 48 bytes of data is  $125 \times 48 = 6,000 \mu\text{s}$ . (The packetization delay for higher bit rate real-time data such as video is proportionately less.) To place this delay in perspective, consider all the delays encountered by a cell as it traverses the ATM network. (See Figure 6.3.)

The cell encounters five types of delay:

1. packetization delay (PD) at the source,
2. transmission and propagation delay (TD),
3. queuing delay (QD) at each switch,
4. a fixed processing delay (FD) at each switch, and
5. a jitter compensation or depacketization delay (DD) at the destination.

We already discussed packetization delay. The propagation delay for electric and optical signals is between 4 and 5  $\mu\text{s}/\text{km}$ , so for a 1,000-km path from source to destination, TD is about 5,000  $\mu\text{s}$ . The transmission delay at a switch is the time needed to transmit one cell, or  $53 \times 8$  bits. If the transmission speed



Assumptions	Delay	Value in $\mu s$
Voice transmission (64 Kbps)	PD = Packetization delay	6,000
Transmission rates = 155 Mbps	TD = Transmission delay (including propagation)	5,000
Length of path = 1,000 km	FD = Fixed processing delay	280
Path goes through 5 nodes	QD = Queuing delay	70
	DD = Depacketization delay	70
	Total delay	11,420
	Delay jitter	70

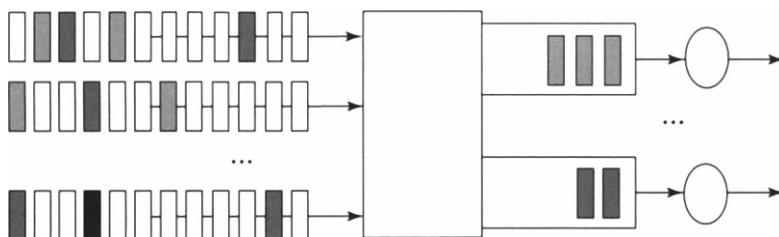
### 6.3 FIGURE

Five types of delays are encountered by ATM cells. The table gives typical values of these delays for a voice conversation.

is 155 Mbps, this time is about  $3 \mu s$ , which is negligible. Hence TD is equal to propagation delay.

A cell suffers queuing delay when there are other cells that arrived earlier or simultaneously and that have not yet been processed by the switch. This delay is random and depends on the traffic load and on the switch architecture. Consider, for example, an output-buffered switch such as that discussed in Chapter 12. Arriving into the buffer of output link 1 (say) are the cells in all the virtual channels that go through link 1. If there are many such virtual channels, and if we assume that the cell streams in these channels are statistically independent, we may approximate the cell arrivals into the output buffer as a Poisson process with rate  $\rho$  per unit of time, where one unit of time is the amount needed to transmit one cell over the output link. See Figure 6.4. (For example, if the link's transmission speed is 155 Mbps, the speed of a SONET STS-3 signal, and a cell contains  $53 \times 8 = 424$  bits, this unit of time is  $424/155 = 2.74 \mu s$ .) Thus the output buffer is modeled as an M/D/1 queue, with service rate equal to one cell per unit time. The average number of cells in the buffer is given by

$$N = \frac{2\rho - \rho^2}{2(1 - \rho)},$$



6.4

FIGURE

In an output buffer switch, the cells that arrive into a buffer are a subset of the cell streams at the input lines of the switch.

where  $\rho$  is the traffic intensity or load or average link utilization. (See the next page for a derivation of this queuing delay formula.) For a load of 80%, this gives  $N = 2.4$  cells, which for a 155-Mbps transmission rate yields QD of about 6.6  $\mu$ s. If a cell goes through 10 such output buffers as it travels from source to destination, QD is about 70  $\mu$ s. (This is average queuing delay; it may be more useful to define QD at the 90 or 99 percentile level, i.e., such that Probability{queuing delay < QD} = 0.9 or 0.99.)

In addition to the queuing delay, a cell undergoes a fixed (almost deterministic) processing delay (FD) as it goes through a switch. This delay is due to the cell being copied into the switch memory one or more times as well as to the time taken for computing the CRC and for translating the VCI into a route through the switch, and so on. The time taken for the memory copies is proportional to the cell size, whereas the time taken for looking up the VCI routing table depends on its size and on how it is organized. Some of these operations may be done in parallel. A reasonable value of FD is about 10 cell times, which for a bit rate of 150 Mbps amounts to 28  $\mu$ s. If the cell goes through 10 switches, total FD is 280  $\mu$ s.

Finally, although voice and other real-time sources generate cells at a fixed rate, those cells arrive at the destination with random intercell delay, called *jitter*. Jitter is mainly due to the random queuing delay. To eliminate jitter, cells arriving at the destination are copied into a buffer that, in turn, is read out at a constant rate. This introduces a depacketization delay (DD) equal to the queuing delay (QD).

The table in Figure 6.3 summarizes the various delays calculated above for a 64-Kbps voice signal transported over a virtual circuit that goes through 10 nodes with 155-Mbps links. We see that for voice, the packetization delay is a significant fraction of the total delay. If the cell size were doubled, so would that packetization delay.

### Queuing Analysis

This section may be skipped by those unfamiliar with probability calculations. We calculate the average delay through an ATM switch. Our model of the switch is an M/D/1 queue. That is, during the  $n$ th cell transmission time, a random number  $A(n)$  of cells arrive at the buffer. The random variables  $A(n)$  are independent and Poisson distributed with mean  $\rho$ . Thus,  $E\{A(n)\} = \rho$ , and one can show from the properties of the Poisson distribution that  $E\{A(n)^2\} = \rho + \rho^2$ . Let  $X_n$  be the number of cells in the buffer at the beginning of the  $n$ th slot time (one slot time is equal to one cell transmission time). Then,

$$X_{n+1} = (X_n - 1)^+ + A_n = X_n + A_n - 1(X_n > 0), \quad n \geq 0, \quad (6.1)$$

where we use the notation that for any number  $z$ ,  $z^+ = \max\{z, 0\}$  and  $1(\cdot)$  is the indicator function, so  $1(z > 0) = 1$  if  $z > 0$ , and 0 otherwise. The term  $(X_n - 1)^+$  accounts for the fact that if  $X_n > 0$ , then one cell will be transmitted, leaving  $(X_n - 1)$  cells in the buffer. Assume that we have reached statistical steady state, so that  $E\{X(n+1)\} = E\{X(n)\}$  and  $E\{X(n+1)^2\} = E\{X(n)^2\}$ . Taking expectations on both sides of (6.1) we get

$$E(X) = E(X) + E(A) - P(X > 0),$$

so

$$P(X > 0) = E(A) =: \rho.$$

Next we square both sides of (6.1) and take expectations to get

$$E(A^2) + \rho + 2\rho E(X) - 2E(X) - 2\rho^2 = 0.$$

Since  $E(A^2) = \rho^2 + \rho$ , we find

$$E(X) = \frac{2\rho - \rho^2}{2(1 - \rho)}. \quad (6.2)$$

#### 6.1.3 Statistical Multiplexing

A virtual circuit specifies a path from source to destination going through several links and switches. Many virtual circuits occupy the same link. A switch has ports terminating several incoming and outgoing links. (A large switch such as in a telephone central office may terminate thousands of links; such a switch is most likely built in modular fashion following the principles presented in Chapter 12. A local area ATM switch, which interconnects terminal equipment

within a local area, may terminate tens of links; that kind of switch is likely to be built in a single stage.)

We now describe the five tasks that a switch carries out. The tasks are demultiplexing, routing through the switch, multiplexing, buffering, and discarding.

The switch demultiplexes the cell stream arriving over each incoming link into “tributary” streams belonging to different virtual channels. The switch does this on a cell-by-cell basis, using the VCI in each cell.

The switch then routes the cell stream in each virtual channel to the appropriate output port. For reasons explained in the next section, the switch may change the VCI assigned to a particular channel. Routing is carried out with the help of a table with entries of the form

( $\text{VCI}_{in}$ , input port,  $\text{VCI}_{out}$ , output port).

(Routers in datagram networks do not have such connection state information. Those routers only have tables with entries [destination address, output port].) An entry in the routing table is created at the time the virtual channel is set up, and it is deleted when that virtual channel is torn down. This routing of the cell stream is internal to the switch; the mechanism that implements routing depends on the switch architecture, as we will see in Chapter 12.

Next, the switch multiplexes the cell streams directed to the same output port. This is statistical multiplexing. (The cell stream in a virtual channel may carry constant bit rate (CBR), variable bit rate (VBR), or message traffic.) The capacity of the output transmission link exceeds the average bit rate of the incoming virtual channels, but it may not exceed their peak rate. Hence, the switch has to buffer excess cells. This need arises when for a short time interval the number of cells to be transferred over an output link exceeds the capacity of that link. Finally, if the buffer is full, the switch must discard cells of lower priority. (ATM specifies two priorities; see section 6.3.)

#### 6.1.4 Allocating Resources

ATM networks offer to transfer cell streams from source to destination, under a range of quality of service, to meet the varying needs of applications, as we saw in section 6.1.1. The service and traffic parameters establish a contract for a particular class of service. The network guarantees the service parameters (such as bounds on delay CTD and loss rate CLR), provided the cell stream emitted by the user conforms to the traffic parameters (such as average rate

SCR and burstiness BT). These parameters are negotiated using established signaling procedures.

In order to meet its obligations under such contracts, the network takes certain actions. It

1. exercises admission control,
2. selects the route (virtual channel path) of admitted connections,
3. allocates bandwidth and buffers separately to each connection,
4. selectively drops low priority cells, and
5. asks sources to limit the cell stream rate (for ABR service).

The ATM Forum has grouped these actions under CAC and UPC procedures. UNI *call admissions control* (CAC) allows the network to grant or deny a connection to a user. *Usage parameter control* or UPC denotes those monitoring and control actions used by the network for compliance to agreed-upon traffic parameters after a connection is granted.

These network decisions are considered in depth in Chapters 8 and 9. Admission control requires the network to determine whether, given existing connections, there are sufficient idle resources to meet the QoS requirements of a new connection request. Once a new connection is admitted, the network must assign to it a virtual circuit with a route that has sufficient resources. (Route selection is discussed in section 6.2.3.) The network must then allocate those resources to the connection so that it can meet the QoS. The first three decisions are taken during the connection setup phase. The last two decisions are taken during the data transfer phase. It happens on occasion that buffers at a switch become full, and some cells must be dropped. If there are cells of different priorities in the buffer, it is preferable to drop those with low priority. Finally, the switch may need to signal to a source to reduce or increase its rate (flow control) depending on how well it conforms to the QoS contract.

In order to implement the routing and resource allocation decisions, the network needs to distinguish between cell streams of different connections. That distinction can be made on the basis of each cell's VCI. Of course, each switch has to create a table of the relation among VCI, its QoS parameters, its route, and its allocated resources and consult this table as needed. The information in the table is said to be implicit in the VCI. (Since referencing the table takes time, we shall see that some of the implicit information is, in fact, explicitly present in each cell of a connection.) By contrast, different cells with the same VCI may have different priorities, and so priority information must be indicated explicitly in each cell. Similarly, the switch may wish to

signal to the source the necessity for flow control, and so explicit provision must be made within the cell for representing this signal. We see how the ATM cell provides this information in section 6.3.

## **6.2** ADDRESSING, SIGNALING, AND ROUTING

In this section we discuss ATM addressing, signaling, and routing in ATM networks. ATM addressing is based on the E.164 international standard for the 15-digit numbering plan used in the public ISDN/telephone system. Since those numbers are too limited to accommodate the growth of private networks, the ATM addressing scheme extends that name space. ATM signaling is an extension of the Q.931 ISDN protocol for signaling between end systems and the public network. ATM routing draws upon concepts used in packet-switched and circuit-switched networks.

### **6.2.1** ATM Addressing

The ATM Forum has defined ATM addressing. An ATM address indicates the location of an ATM interface in the network topology. This means that ATM addresses are not portable. (Anycast service, discussed below, is an exception.) The prefix of an address is associated with a group of interfaces with the same prefix. Prefixes are used in call routing tables.

Each ATM system is assigned an address independent of the higher protocol addresses (such as IP addresses) that the system supports. The decoupling afforded by this *overlay model* allows the higher-level protocols to be developed independently of the ATM protocols. Hence all protocols operating over an ATM subnet require an address-resolution protocol that maps the higher protocol address into the corresponding ATM address. For example, to send an IP packet to a given IP address over an ATM network, routers use the address-resolution protocol to determine the ATM address of the destination. The routing algorithm then sets up the connection to the destination. We discuss IP over ATM in more detail later.

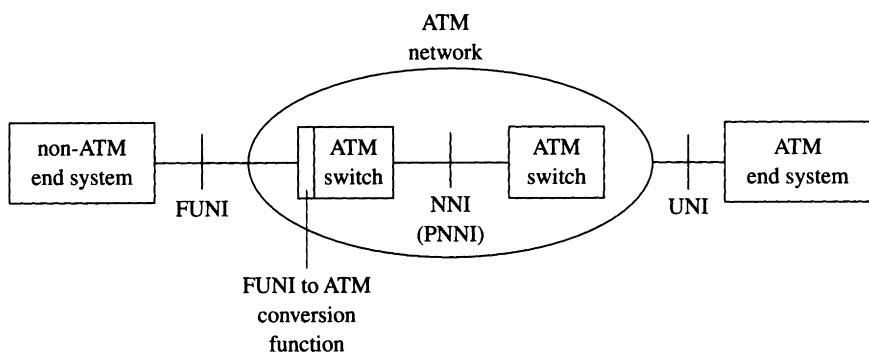
An ATM address is 20 bytes long. The last 7 bytes contain a 48-bit MAC address, followed by a 1-byte select field that has no network significance. The prefix may contain a 15-digit E.164 number (specified by ITU-T for establishing ISDN and telephony calls) or another hierarchical address. The ATM Forum defines an address registration mechanism using the ILMI (Integrated Local

Management Interface) protocol, discussed in section 6.5.3. This allows an end system to inform an ATM switch to which it is attached of its MAC address and get back its full ATM address.

An ATM Group Address represents a collection of ATM end systems. An end system may join or leave a group by registering a group address using the ILMI. The only ATM Forum-defined service using group addressing is the *anycast* service. Anycast service, defined in UNI 4.0 and PNNI 1.0, allows multiple systems to register as a server for some service (e.g., printer server, file server). A well-known anycast address can be used to route a request to a node providing a particular service without explicitly identifying the node. A call made to an anycast address is routed to the “nearest” end system that registered itself with the network to provide the associated service. Anycast is a useful mechanism for network configuration and operation since it precludes the need for manual configuration or service locations protocols.

### 6.2.2 Signaling

An ATM network consists of ATM switches connected by point-to-point links or interfaces. The ATM Forum specifies several interfaces, as seen in Figure 6.5. The two most important are the *user-network interface* or UNI and the *network-node* or *network-network interface* or NNI. A UNI connects ATM end systems (hosts, routers) to an ATM switch, and an NNI connect two ATM switches. The formats of the cells flowing across these interfaces are described in section 6.3.



**FIGURE**  
6.5

UNI is the interface between an ATM end system and an ATM switch, NNI is the interface between two ATM switches, and FUNI is the frame interface between a non-ATM end system and an ATM switch.

UNI is further distinguished by private or public UNI, depending on whether the end system is connected to a private or public network, and similarly for NNI. PNNI stands for *private network-network* or *node-network interface*.

The public network-network interface is specified in B-ICI or *BISDN Inter Carrier Interface*. It currently provides only for PVC (permanent virtual connections), although work has started for switched virtual connections (SVCs). It is possible that the powerful facilities of route discovery and aggregation in PNNI may be imported into the public networks.

The frame-based user-network interface or FUNI permits a non-ATM legacy system to connect to an ATM, exchanging frames rather than ATM cells. Frames of size up to 64 KB may be supported at a speed of  $n \times 64$  Kbps up to 1.5 Mbps, but ongoing work may increase this speed. The ATM switch carries out the FUNI to UNI conversion function via AAL5. FUNI brings most of the UNI control signaling to non-ATM end systems. The ATM Forum and ITU-T have also specified other interfaces between non-ATM legacy systems and ATM switches.

UNI 4.0 and PNNI 1.0 specify the signaling protocols used across UNI and NNI. The PNNI signaling protocols are based on UNI 4.0, and so only the latter is described here. PNNI signaling makes use of the information gathered by PNNI routing.

UNI specifies the signaling procedures for dynamically establishing, maintaining, and clearing ATM connections at the UNI interface.

There are two basic types of connections: point-to-point and point-to-multipoint. Point-to-point connections may be unidirectional or bidirectional. Point-to-multipoint connections connect a single root node to multiple destinations called *leaves*. These connections are unidirectional: a root can transmit to leaves, but leaves cannot transmit to the root. Cell replication within the network is done by the ATM switches where the connection splits into two or more branches.

Signaling procedures are defined in terms of message types, information elements (IE) that carry data such as addresses and QoS requirements of an ATM connection, and state machines that describe the procedures. (QoS requirements are expressed as values of the parameters in Table 6.1.) Signaling requests are carried across the UNI in a well-known default connection (VPI = 0, VCI = 5). The specification is based on Q.2931, a public network signaling protocol developed by ITU-T, which, in turn, was based on the Q.931 protocol developed for ISDN.

A source end system attempting to set up a point-to-point connection creates and sends into the network, across its UNI, a Setup message, containing the destination address, desired traffic and QoS parameters, IEs defining desired

higher-layer protocol bindings, and so on. This Setup message is sent to the ingress switch, which responds with a local Call Proceeding acknowledgment. The switch then invokes the PNNI routing protocol to propagate the signaling request through switches across the network, setting up connection identifiers (VPI/VCI) along the way, to the egress switch to which the destination end system is attached.

The egress switch forwards the Setup message to the destination end station. The latter may either accept or reject the connection request. In the former case, it returns a Connect message, back through the network, along the same path, to the requesting source end system. Once the source end system receives and acknowledges the Connect message, either node can then start transmitting data on the connection. If the destination end system rejects the connection request, it returns a Release message, which is also sent back to the source end system, clearing the connection (e.g., any allocated connection identifiers) as it proceeds. Release messages are also used by either of the end systems, or by the network, to clear an established connection. The data flows along the same path traveled by the connection request.

It is possible that a switch along the path rejects a request. This may happen, for instance, because the switch realizes that the resources specified in the request are not available. The switch then sends a reject message along the reverse path to the ingress switch. That switch may then try to repeat the procedure to find an alternate path. This procedure is called *crankback* by the ATM Forum. It is similar to the dynamic alternate routing of circuit-switched networks (see section 8.2.2). The ingress switch may eventually reject the connection request from the source. The procedure followed by a switch to grant or reject a request is called connection admission control or CAC.

A multipoint connection is initiated by a Setup message from the root node to a single leaf node. The root can then send Add Party or Drop Party messages to add or delete leaves in an existing connection. There is also a Leaf-Initiated Join capability in which a leaf node can join an existing connection with or without the intervention of the root node.

### 6.2.3 PNNI Routing

The ATM Forum's Private Network-Network Interface (PNNI) version 1.0 includes two classes of protocols. There is a protocol for distributing topology information between switches and clusters of switches. This information is used to compute paths or routes through the specification network. A hierarchy mechanism ensures that this protocol scales well for large worldwide ATM networks. The second protocol is for signaling. As noted above, PNNI

signaling is based on UNI signaling, with mechanisms to support source routing, crankback, and alternate routing of call setup in case of connection setup failure.

We will first discuss topology information and routing assuming a flat or nonhierarchical network, and then discuss the hierarchy mechanism.

PNNI routing uses a generalization of the link state (Dijkstra's) algorithm discussed in section 4.3. The generalization consists in the fact that a link's state is not a single metric representing, say, delay, but a vector of "topology state parameters," including "link state parameters," which describe the characteristics of logical links, and "nodal state parameters," which describe the characteristics of nodes.

Topology state parameters are classified as attributes or metrics. An attribute is considered individually when making routing decisions. For example, a security "nodal attribute" could cause a proposed path to be refused. The capacity of a link is also an attribute: a connection request for a specified bandwidth (SCR or sustained cell rate) must be met by each link along the route. A metric, on the other hand, is a parameter whose effect is cumulative along a path. For example, a delay metric adds up as one progresses along a given path.

The PNNI specification designates Routing Control Channels (particular VPI/VCI) for the exchange of PNNI routing packets (Hellos, PTSP, PTSE acks, etc.) between switches.

Nodes in the network exchange Hello packets with their immediate neighbors and thereby determine their local state information. This state information includes the identity and peer group membership (described below) of the node's immediate neighbors, and the status of its links to the neighbors. Each node then bundles its state information in *PNNI Topology State Elements* (PTSEs) reliably flooded throughout the peer group. (PTSEs are encapsulated in *PNNI Topology State Packets* or PTSPs.) In this way, each node obtains the link and node state parameters of the entire network. This information is used by the ingress switch to compute a path that can satisfy a connection request received from a source end station.

When the ingress switch determines the complete path, the path is included in the connection request as a *Designated Transit List* or DTL. Thus PNNI uses source routing. The PNNI does not specify how to compute the source route. We formulate a mathematical problem that such a computation should solve.

The topology information at a switch can be represented as a graph whose nodes are the ATM switches and whose edges are the links. Each node  $i$  is labeled with its node attribute parameter  $\theta(i)$ , which determines whether or

not a requested connection can traverse the node. Each link  $(i, j)$  is labeled with a vector  $d(i, j) = (d_1(i, j), \dots, d_K(i, j))$  of the  $K$  link state metrics. (Link attributes are ignored: they can be treated like node attributes.)

A node parameter  $\theta$  is a predicate on connection requests, so that a particular request can be routed through the node only if it satisfies the predicate. The link metrics represent quality characteristics such as delay, or jitter, so that the characteristics of the path or route are the sum of the characteristics of the links along the path.

A connection setup request is a triple

$$R = (i, j, D = (D_1, \dots, D_K)),$$

where  $i$  is the source node,  $j$  is the destination node, and  $(D_1, \dots, D_K)$  specify the connection's QoS requirements.

A *feasible path* is a sequence of nodes  $i = i_0, \dots, i_N = j$  such that

$$R \text{ satisfies } \theta(i_n), \quad n = 0, \dots, N \text{ and} \tag{6.3}$$

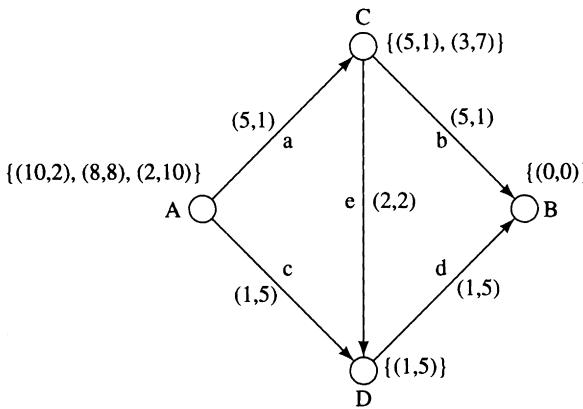
$$\sum_{n=1}^N d_k(i_{n-1}, i_n) \leq D_k, \quad k = 1, \dots, K. \tag{6.4}$$

Requirement (6.3) states that the nodes along the path must accept the connection, so this requirement effectively deletes nodes from the graph that cannot accept the request. We assume henceforth that those nodes have been deleted and so we focus on (6.4). It states that the quality characteristics along the path must meet the requirements of the request. So the routing task is to find a path that satisfies each of the  $K$  QoS requirements (6.4).

Observe that if the link state is a single number ( $K = 1$ ), one can use the shortest-path algorithms of section 4.3. But when  $K > 1$ , the computation of feasible paths is more complex. We explain why this is so in the example of the five-link network of Figure 6.6. There are no node parameters. Each link has two metrics as shown. Thus for link  $a$  the values of the link state parameters are  $(d_1(a) = 5, d_2(a) = 1)$ , and similarly for the other links.

Consider three requests for a connection from  $A$  to  $B$ . The first request is  $(D_1, D_2) = (10, 2)$ . The only path that can satisfy this request comprises the links  $a, b$ . The request  $(2, 10)$  is met only by the path  $c, d$ . Lastly, the request  $(8, 8)$  is met only by the path  $a, e, d$ .

In the case of a single metric, there is a shortest-path tree rooted at the destination node  $B$ . This tree can be obtained using, for example, the Bellman-Ford



6.6

A five-link network with two link state metrics.

FIGURE

algorithm of section 4.3. In that algorithm one proceeds upstream, estimating the length of the shortest path to the destination from node  $i$  by

$$L(i) = \min_j [d(i, j) + L(j)]. \quad (6.5)$$

The algorithm starts with the initial estimate  $L(B) = 0$  and  $L(i) = \infty$ , for all other nodes.

For our case where  $d$  is a vector of metrics, this algorithm can be modified as follows. We estimate a *set* of metric vectors  $L(i) \subset R^K$  by:

$$L(i) = \mu(\{d(i, j) + l(j) \mid j = 1, \dots, n; \quad l(j) \in L(j)\}), \quad (6.6)$$

where for a set  $L \subset R^K$ ,  $\mu(L)$  is the subset of noninferior metric vectors:  $l^* \in L$  is noninferior if there is no other  $l \in L$  with  $l_k \leq l_k^*$ ,  $k = 1, \dots, K$ . The algorithm (6.6) starts with the initial estimate  $L(B) = \{(0, \dots, 0)\}$ , and  $L(i) = \{(\infty, \dots, \infty)\}$  for all other nodes  $i$ .

Figure 6.6 shows the final estimates obtained from this algorithm. From the lists displayed at each node we can determine whether there is a feasible path from any source to the destination  $B$ . For instance, the request for a connection from  $C$  to  $B$  with characteristics  $(4, 7)$  is feasible since  $(3, 7) \leq (4, 7)$  and  $(3, 7) \in L(C)$ . On the other hand, the request for a connection from  $A$  to  $B$  with characteristics  $(7, 9)$  is not feasible.

In the case of a single metric, the routing table at any node has a simple structure. It contains, for each destination, the estimate of the minimum

distance and the next node along the shortest path. In the case of multiple metrics, the routing table must now store, for each destination, the set of noninferior metric vectors, and for each such vector the next node along the corresponding path.

The algorithm suggested above is not scalable. To be scalable, a routing algorithm must be hierarchical. The PNNI routing algorithm defines a hierarchy of nodes. At each level, nodes are clustered in peer groups. The nodes of the same peer group elect a peer group leader, which maintains an aggregate description of the group. This aggregate description indicates the characteristics of the service across the group. The precise procedure for deriving such an aggregate description is not fully specified. The group members also maintain routing tables to reach each other and their peer group leader. The peer group leaders exchange information to identify their neighbors and to calculate routing tables to reach one another. At the next level up, these peer group leaders are then clustered into a parent peer group, and so on. The addresses of the nodes are designed to identify their group memberships, as the telephone numbering does (country code, area code, zone, number in zone). Note that when a group of nodes is aggregated, there is a loss of topology information. The feasible routes constructed on the basis of the aggregated information might turn out to be infeasible. The connection request will then be rejected, and the ingress switch will resort to crankback.

To route from A to B, the routing algorithm can then look at the addresses to find out the group leaders that should be involved. Thus, if the address of A is XYZ and that of B is XVW, then A and B belong to the *same group or groups of nodes*. The routing will then go from A to its peer group leader, say C, to the parent peer group leader of C, say D, to the peer group leader of B, say E, to B itself. The routing table at A specifies how to get to C. The table at C specifies how to get to E, and the table at E indicates how to reach B. At each level, the routing is decided by an algorithm, possibly of the kind suggested above, that incorporates QoS characteristics.

If the connection must go through a connectionless public service, such as Frame Relay or SMDS, then it becomes almost impossible to guarantee the quality of service of the connection. This is the case even though one can transmit the desired characteristics of the connection across the public service to other private ATM networks.

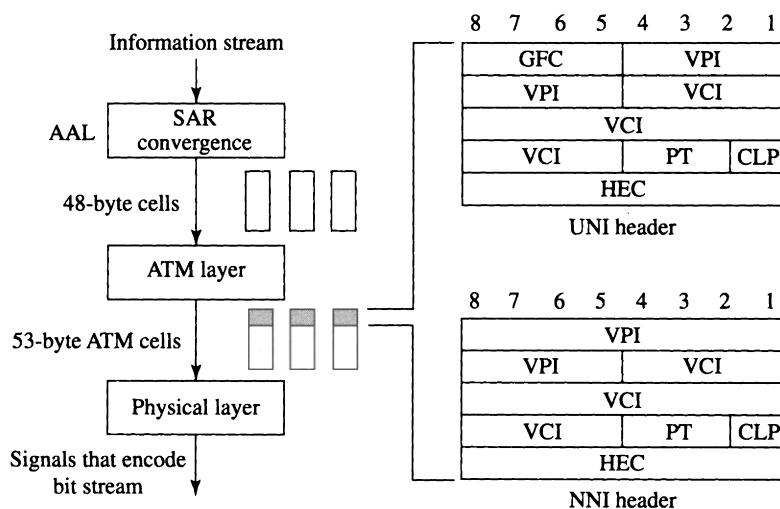
It is interesting to note how earlier concepts are borrowed and improved. PNNI extends the shortest-path protocols to include QoS and source routing. In turn PNNI innovations of source routing (such as DTL) are adopted by MPLS (see section 4.6).

## 6.3 ATM HEADER STRUCTURE

In this section we will examine the ATM cell structure, and we will see how an ATM switch can obtain the information it needs from the cell header. As indicated in Figure 6.7 the AAL (ATM adaptation layer) produces a data stream of 48-byte cells or PDUs (protocol data units). (We will see later the function of the AAL layer.) The ATM layer adds a 5-byte header and forwards the 53-byte cell to the physical layer. The 53-byte cell is converted into a serial bit stream by reading the cell from left to right (most significant bit first) and top to bottom (byte 1 first).

The figure shows the header structure for both the user-network interface (UNI) and the network-network interface (NNI). The abbreviations used in the figure are

- ◆ GFC, generic flow control,
- ◆ VPI, virtual path identifier,
- ◆ VCI, virtual channel identifier,
- ◆ PT, payload type,



6.7

FIGURE

The figure shows the headers of ATM cells across the user-network interface (UNI) and across a network-network interface (NNI).

- ◆ CLP, cell loss priority, and
- ◆ HEC, header error control.

The only difference between the two headers is that the UNI header has a 4-bit field that can be used to signal to the user the need for flow control. The NNI uses those bits to expand the VPI field.

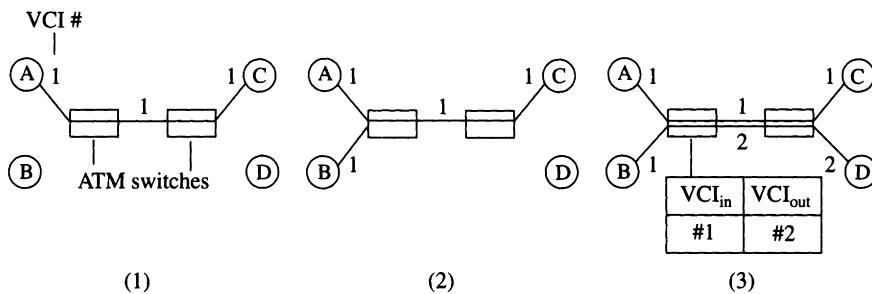
### 6.3.1 VCI and VPI

Consider first the 8-bit VPI and the 16-bit VCI. The VPI/VCI combination is the connection identifier. A virtual path identifier (VPI) groups many virtual channel identifiers (VCI). ATM allows two levels of multiplexing: virtual channels may be multiplexed into the same virtual path, and virtual paths may be multiplexed into the same link.

The most important feature is that the VCI is *local* to each VPI, and the VPI is *local* to each link. More precisely, different simultaneous connections that share the same VPI must have different VCIs, and virtual paths that share the same link must have different VPIs. But connections on different VPIs may share the *same* VCI, and virtual paths on different links may have the *same* VPI.

In the example below, suppose that the entire link is devoted to a single VPI, so VCIs are local to a link. Because VCIs are local to a link, connections coming into a switch from different switches (or sources) may have the same VCI. However, if these connections share the same outgoing link, they must be assigned different VCIs on that link. This works as in Figure 6.8, which shows four user nodes (A, B, C, D) and two switches. Initially there is a connection over VCI #1 from A to C. At some later time there is a request to establish a connection from B to D. Since B is not currently using VCI #1, it assigns that VCI to the connection. When the packet establishing this connection reaches the first switch, that switch knows that VCI #1 is assigned to another connection that shares a link with the proposed connection. The switch therefore changes the VCI from #1 to #2, noting the change in a table. (During the data transfer phase, this switch must change the VCI on each cell on the B-D connection from #1 to #2.) Strictly speaking, the VPI/VCI is assigned by the network and not by the user as suggested in the discussion.

The 16-bit VCI field permits 64,000 simultaneous connections through each link, and since the same identifier may be reused by connections with disjoint paths, the network can support orders of magnitude of more simultaneous connections.

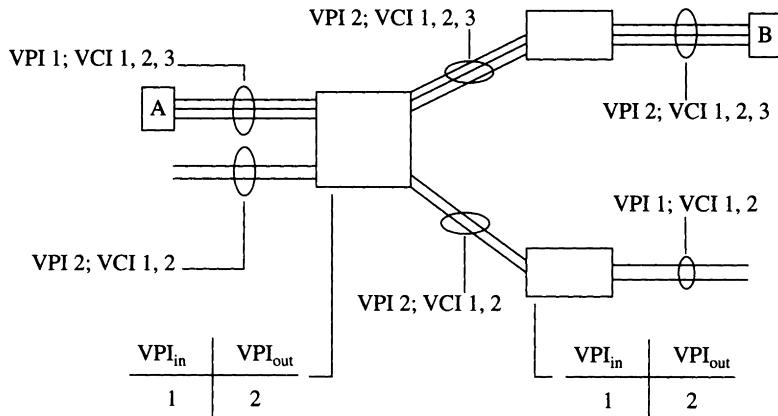


6.8

FIGURE

Virtual channel identifiers must be unique per connection on any given link. When the network sets up the second virtual channel, it assigns it a different VCI on the link that the second virtual channel shares with the first. (1) VCI #1 from A to C has been set up. (2) Request from B for connection to D is initially assigned VCI #1. (3) Because of the common link, the new connection is given VCI #2. The switch creates and updates the VCI translation table.

Some purposes may be better served by treating several virtual channels together as a group. Suppose, for example, as shown in Figure 6.9, that a user wishes to establish three virtual channels from A to B. It would then make sense to give these channels the same path and permit the user to assign VCIs to those channels arbitrarily. That is the function of the VPI. Virtual channels



6.9

FIGURE

A virtual path is a group of virtual channels that the network routes together.

with the same VPI form a group. They are assigned the same path and are switched together, that is, routing and switching decisions are based only on the VPI. More interestingly, the bandwidth and buffer resources allocated to a VPI may be assigned statically (at the time of service subscription) and shared only by virtual channels with that VPI. This use of VPIs permits the creation of *virtual private networks*: a multilocation firm can rent several virtual paths to form its own private network whose resources are then shared by its virtual channels.

From Figure 6.7 we note that the UNI header has an 8-bit VPI field, while the NNI has an additional 4 bits. The network may use these additional bits to create certain fixed routes, similar to what is done in today's long-distance telephone networks. This may allow resources allocated to virtual paths to be statically assigned and shared dynamically among component virtual channels. The use of VPIs may speed up processing, since switches only need to consult a table with entries indexed by shorter VPIs.

### 6.3.2 Other Fields

The 4-bit GFC (generic flow control) may be used by the network to signal to a user the need for momentary changes in the instantaneous cell stream rate. The functionality of the GFC has not yet been established. Since GFC is not present in the NNI header, it can be used only to control the flow at the UNI interface.

The 3-bit PT (payload type) permits networks to distinguish among different types of information. If the first bit is 0 (PT = 0xx), it indicates user information. In that case the second bit indicates congestion not experienced (PT = 00x) or congestion experienced (PT = 01x). The third bit is used to distinguish two types of payload, depending on the context. For example, it is used to indicate the end of a packet in AAL 5 (see section 6.4).

PT = 100 or 101 is used for maintenance and for network equipment to introduce and remove special cells that are routed as ordinary cells but that carry special information for control purposes. PT = 110 is for resource management cells.

The 1-bit CLP (cell loss priority) distinguishes between cells that the network may not discard (CLP = 0) and cells that it may discard (CLP = 1) if necessary. Of course, the network may introduce a low-priority service (as a possible QoS) at the level of a connection—all cells in such a connection are subject to discard. Since that information is implicit in the VCI, the CLP field is not needed, although it may be used to make the information explicit. The CLP field is needed, however, if different loss priority cells are present

in the same connection. One example would be if voice were encoded into equal numbers of high and low order bits and sent over cells that alternately carry the high and low order bits. The low order bit cells would be assigned a CLP = 1, and would be subject to discard, while the cells with high order bits would be assigned CLP = 0 and would not be discarded. By encoding voice in this way, the statistical multiplexing gain can be increased considerably. Also, if a border switch notices that cells do not conform to the service parameters, then that switch can set CLP = 1 in those cells so that these nonconformant cells are eligible for discard. The border switch detects such nonconformant cells by using the GCRA with the traffic parameters, as we explain in section 8.4.2.

Finally, the 8-bit HEC (header error control) field is equal to the sum of the byte 0101'0101 and the CRC (cyclic redundancy check) calculated over the rest of the header with the generator polynomial 1'0000'0111. (The functioning of the CRC is discussed in section 2.6.3.) As explained in section 6.1.2, the HEC is also used to delineate the cell boundary. The HEC can correct single bit errors and detect multiple bit errors. The error-control algorithm has two states: error detection (D) and error correction (C). The algorithm is initially in state C. If the algorithm is in state C and detects a single bit error, it corrects the error and moves to state D. If it detects a multiple bit error when in state C, the algorithm discards the cell and moves to state D. The algorithm discards cells that contain errors when in state D and remains in state D. The algorithm moves from state D to state C when it gets a cell without error.

### 6.3.3 Reserved VCI/VPI

Some VCI/VPI combinations are reserved. One set of combinations transports so-called unassigned cells. These may be introduced by the transport layer at the source, but they do not contain user information. These cells are removed by the destination transport layer; they are not forwarded to the application layer.

Other combinations are reserved for UNI and PNNI signaling and for ILMI messages.

Sets of VCI/VPI combinations are reserved for cells that can be introduced and removed by the physical layer. These cells are invisible to the ATM layer. One set is assigned to "idle" cells that may be introduced periodically to help synchronization (cell boundary recovery). If the bit stream generated by the physical layer is synchronous (as in SONET), then idle cells may be used to fill up a frame whenever there is no user traffic to be sent. Another set of VCI/VPI combinations is assigned for OAM (organization, administration,

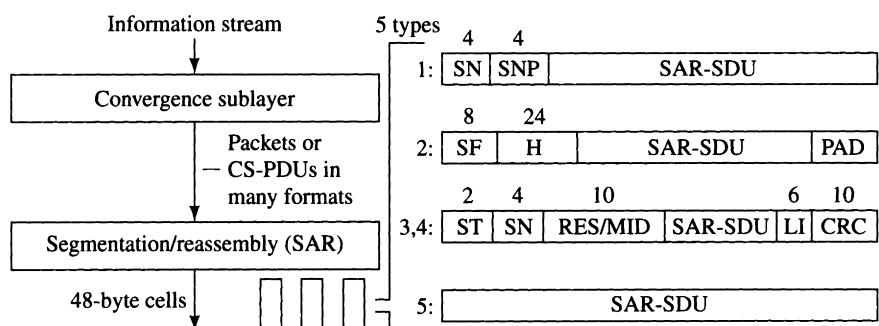
and management) functions at the physical layer. In essence, these form connections dedicated to assist in various monitoring and physical layer control functions.

## **6.4 ATM ADAPTATION LAYER**

As shown on Figure 6.10, the network converts the information stream into a stream of 48-byte data cells. This conversion is performed by the *ATM adaptation layer* or AAL. The AAL is divided into two sublayers: the CS or convergence sublayer and the SAR or segmentation and reassembly sublayer.

The CS converts the information stream into four types of packet streams, called AAL Type 1, Type 2, Type 3/4, and Type 5. The packet formats match the requirements of the information stream, classified into five types of traffic: constant bit rate-real time, variable bit rate-real time, connection-oriented packet streams, datagrams, and IP packets, as indicated in Table 6.1. Five AAL types of packets were originally envisioned. AAL Type 2 was recently specified for use in narrowband services.

The individual packets are called SDUs (service data units), following OSI terminology. Since a user information stream may be encoded into packet streams in many different ways (e.g., video may be encoded into a constant or variable bit rate stream), and since some of this encoding information must be included in the SDU, CS tasks are likely to be application-dependent. Hence



**6.10**  
**FIGURE**

The ATM adaptation layer (AAL) converts the information stream into 48-byte cells. The AAL is decomposed into the convergence sublayer and the segmentation/reassembly sublayer.

CS is further subdivided into the upper, service-specific or SSCS sublayer and the lower, common part or CPCS sublayer.

By contrast, SAR tasks are quite standard, depending only on traffic type. The SAR must segment the SDUs received from the CS, add the necessary overhead, and convert the result into 48-byte cells or PDUs (protocol data units), which are then handed over to the ATM layer. As we will explain, the overhead is needed by the SAR at the destination to reassemble the SDUs from the 48-byte cells.

We describe each type, its typical intended applications, and the corresponding SAR functions.

#### 6.4.1

#### Type 1

This traffic is generated at a constant bit rate, and it is required to be delivered at the same rate (with a fixed delay). Intended applications are voice and constant bit rate video or audio. Since the cells may suffer variable delays, the CS at the destination must compensate for those delays. This is done in one of two ways. Incoming cells are buffered at the destination and the buffer is read out at a fixed rate.

Alternatively, the CS at the source inserts an explicit time stamp into the packet stream. The destination CS extracts this time stamp and attempts to read cells with a constant latency. (A cell that arrives with a delay that exceeds this latency may be discarded.) The source and destination CS sublayers must agree on the scheme to be used for timing information. That agreement must be reached during the connection setup phase.

Figure 6.10 displays the structure of the 48-byte PDU for Type 1 traffic. The SAR sublayer takes the (periodic) packet stream generated by the CS sublayer, segments it into 47-byte SDUs, and prepends to each SDU a 4-bit sequence number (SN) protected by a 4-bit sequence number protection (SNP) field. The SNP corrects single bit errors and detects multiple bit errors in the sequence number. The eight possible sequence numbers permit the SAR sublayer at the destination to determine if fewer than eight consecutive SDUs are lost. (The source and destination must agree on what to do when cell loss is discovered.)

#### 6.4.2

#### Type 2

AAL Type 2 was recently specified. It is intended to serve the need to transport low bit rate traffic such as compressed or uncompressed voice, compressed video, and fax. It allows multiplexing more than one AAL 2 traffic stream over the same connection. AAL 2 permits a user packet to be split across two cells.

The PDU has three parts. The 8-bit start field (SF) includes a 6-bit offset field (OSF), which indicates the number of bytes between the start field and the payload, or between the start field and the next packet in case the payload contains the end of one user packet and the beginning of the next packet. The next field is the 24-bit header (H) of the SAR-SDU. The header includes the channel ID (CID) for use in multiplexing several streams, and the length indicator (LI) for variable-length payload. The SAR-SDU payload can have variable length and may have multiple SDUs from different AAL 2 streams. The final field comprises padding bytes, if needed, to fill up the 48 bytes.

AAL 2 may be used to support voice over ATM. The specification does not define the timing relationship between sending and receiving applications, or how to compensate for jitter. That is left to applications. For instance, the Real Time Protocol (see Chapter 4) provides a time stamp.

### 6.4.3 Type 3/4

The original standard specified two types: AAL 3 for connection-oriented streams, and AAL 4 for connectionless messages. The two types are now combined into a single type, AAL 3/4, intended for internetworking SMDS and MAN networks over ATM. AAL3/4 accepts CS-PDUs of up to 64 KB, which are segmented into 53-byte cells for transmission. Two types of data transfer are supported: a message-mode service which allows a single SDU and a stream-mode in which one or more fixed-size SDUs are transported.

Two kinds of error quality are envisaged. In the first kind the CS sublayer guarantees error-free delivery. In the second kind no such guarantee is made. Error-free delivery is ensured by the destination CS requesting retransmission when an error is detected. To ensure reconstruction of the packets and to detect errors, the overhead fields appended by the SAR sublayer are as in Figure 6.10. The 2-bit ST or segment type is set as follows: ST = 10 for BOM, ST = 00 for COM, ST = 01 for EOM, and ST = 11 if the packet fits inside a single cell. The 4-bit SN (sequence number) is used as before to detect loss or cells inserted by network routing errors. The 10-bit MID field is explained below. The 6-bit LI field is needed when the cell is only partially filled. Finally the 10-bit CRC is a check over the SAR-SDU used to detect errors in the CS SDUs.

It would be common for a user to multiplex several different datagrams over the same virtual channel connection. The MID (multiplexing identifier) field can be used to distinguish among the cells originating from different datagrams. (This is similar to the use of the MID or message identifier field in SMDS.)

#### 6.4.4

#### Type 5

This traffic carries IP packets. The frame structure of Type 5 eliminates the overhead present in Type 3/4. The IP packet, up to 64K bytes, is packaged into a CS packet that contains a length indicator and a 32-bit CRC calculated over the complete CS packet with the generator

1'0000'0100'1100'0001'0001'1101'1011'0111.

The CS packet contains a padding field so that the length of this CS packet is an exact multiple of 48 bytes. The CS packet is sent as a sequence of SAR packets of AAL Type 5. The PT of the ATM cell header indicates whether the cell is the last one of the CS packet or not.

---

## 6.5

## MANAGEMENT AND CONTROL

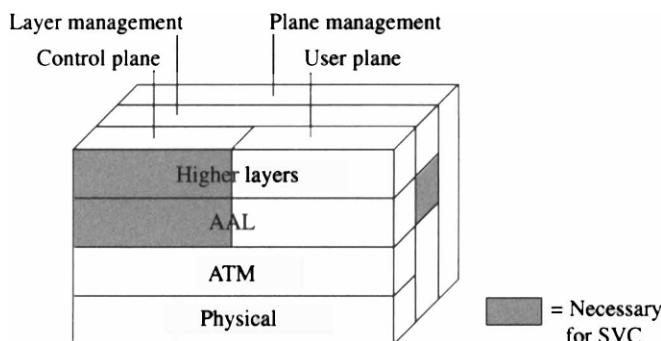
A very important feature of ATM networks is that they can make a number of management and control decisions to discriminate among connections and to provide the variety of QoS that different applications need. The decisions are divided into three groups. When a request is made for a connection with a particular QoS, the network must determine whether to accept or reject the request, depending on the resources then available. (Recall that QoS involves three sets of parameters: delay, cell loss, and source traffic rate.) If the resources are insufficient to meet the request, the network may negotiate with the user the traffic parameters in the requested service class.

Once the connection is admitted, the network must assign a route or path to the virtual channel that carries the connection. It must inform the switches and other network elements along the path that this virtual channel must be allocated certain resources so that the agreed-on QoS is met.

Lastly, the network must monitor the data transfer to make sure that the source also conforms to the QoS specification and to drop its cells as appropriate. (This is called *traffic policing*.) The network may also ask a source to slow down its transmissions.

In addition, the network carries a number of information flows to monitor its operations and to detect and identify the location of congested or failed devices.

The BISDN standard so far is silent about how these decisions are to be carried out. (However, recent ITU recommendations deal with OAM, performance monitoring and protection switching at the ATM layer to complement



**6.11**  
**FIGURE**

The BISDN model layer arrangement of network functions, including the operation and management functions.

SONET protection switching.) We shall discuss potential solutions in Chapter 8. The ATM Forum specifies frame formats that the network should use to carry its monitoring information and to interact with users. We review these next.

The network uses operation and maintenance information flows for the following functions:

- ◆ fault management,
- ◆ traffic and congestion control,
- ◆ network status monitoring and configuration, and
- ◆ user/network signaling.

These functions, like the other network functions, are organized into layers, called the *BISDN reference model*.

Figure 6.11 shows the layer arrangement of all network functions, including those of operation and management. The layers in the *user plane* comprise the functions required for the transmission of user information. For instance, for an Internet Protocol over ATM, these layers could be HTTP/TCP/IP/AAL5.

The layers in the *control plane* are the functions needed to set up, supervise, and release a virtual circuit connection. These functions, implemented by signaling protocols such as PNNI, are needed only for switched virtual connections and are absent in a network that implements only permanent virtual connections. (In a permanent virtual circuit connection, the path or route assigned to a source and destination and the VCI for that route are fixed.)

The *layer management plane* contains management functions specific to individual layers. Layer management also handles the operations and main-

tenance flows specific to each layer. The protocols used for these functions include ILMI and SNMP.

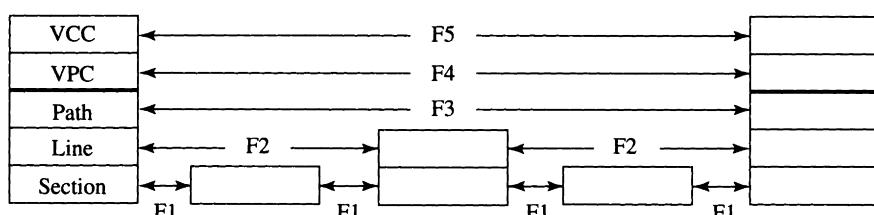
Finally, *plane management* consists of the functions that supervise the operations of the whole network. Plane management has no layered structure.

### **6.5.1 Fault Management**

Consider a virtual circuit connection over an ATM network and assume that the connection is implemented by a SONET network. We know from section 5.2 that SONET establishes transmission *paths* for the ATM layer. The transmission is over optical fibers. The transmitters in SONET are all synchronized to the same master clock. This synchronization enables the time-division multiplexing of different bit streams. This multiplexing is done byte-by-byte.

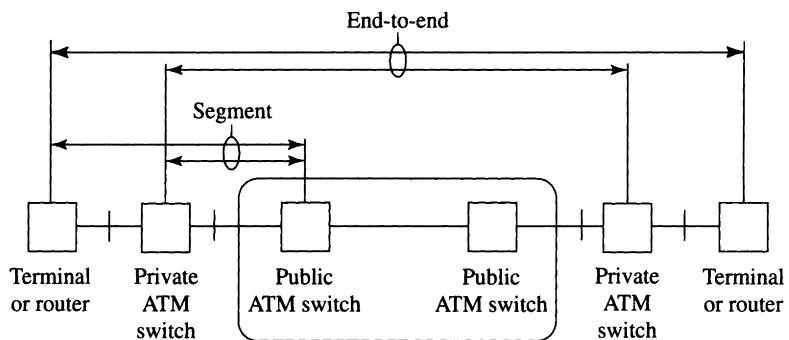
The physical layer (SONET) is decomposed into three sublayers: section, line, and path. The section layer transmits bits between any two devices where light is converted back into electronic signals or conversely. For instance, there is a section between two successive regenerators or between a regenerator and a multiplexer. The line layer transports bits between multiplexers where SONET signals are added to or dropped from the transmission. Finally, the path layer transports user information. Thus, a path goes across a number of lines (or links) that are switched by the SONET demultiplexers and multiplexers, and a line consists of a number of sections. Each layer inserts and strips its own overhead information, which it uses to monitor the transmission functions for which it is responsible. (See Figure 6.12.)

Each of the three sublayers uses overhead bytes in the SONET frames to supervise its operations. The overhead bytes are said to carry a *flow* of operation and maintenance information. The flow carried by the section overhead bytes is called F1. The flow carried by the line and path overhead bytes are F2



6.12

## Operation and maintenance flows for a virtual circuit connection over SONET.

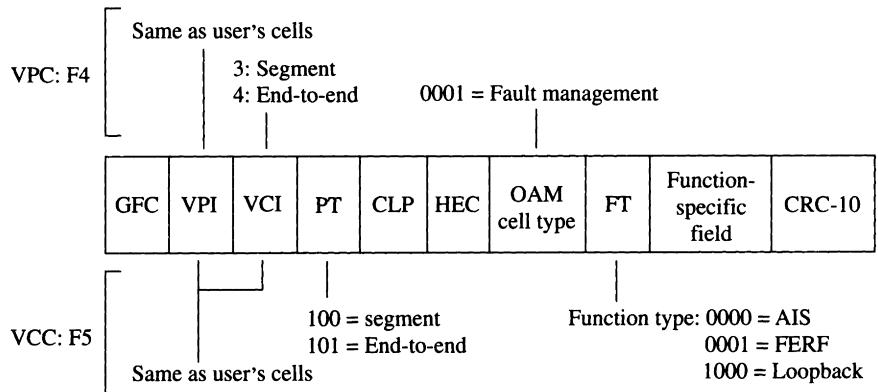


**6.13  
FIGURE**

A segment indicates a connection across the user-network interface. An end-to-end connection is between the source and destination user equipment.

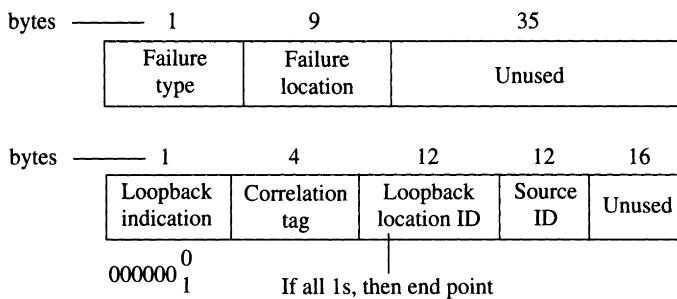
and F3, respectively. The virtual circuit connection is carried by a virtual path connection. Accordingly, the network uses a flow of cells to supervise the virtual path connection and a flow of cells to supervise the virtual circuit connection. These two flows are called F4 and F5, respectively.

The format of the F4 and F5 cells depends on whether the cells monitor the segment across the user-network interface or the end-to-end connection (see Figure 6.13). The cell formats are shown in Figure 6.14. Note that the F5 cells have the same VPI/VCI as the user cells of the connection they monitor.



**6.14  
FIGURE**

Format of OAM cells.



6.15

FIGURE

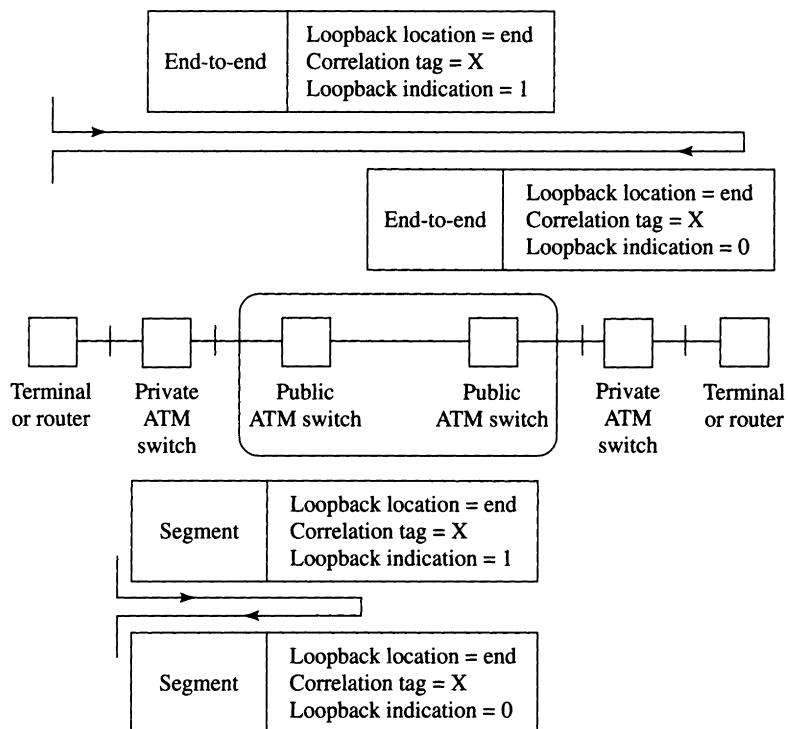
Function-specific fields in AIS and FERF cells (above) and in loopback cells (below).

The F5 cells are distinguished from the user cells by the PT field. Similarly, the F4 cells have the same VPI as the user cells and are distinguished by their VCI.

The main function of the OAM cells is to detect and manage faults. Fault-management OAM cells have the leading 4 bits of the cell payload set to 0001. The next 4 bits, the function type (FT) field, indicate the type of function performed by the cell: alarm indication signal (AIS), signaled by FT = 0000; far end receive failure (FERF), signaled by FT = 0001; and loopback cell, signaled by FT = 1000. The AIS cells are sent along the VPC (virtual path connection) or VCC (virtual circuit connection) by a network device that detects an error condition along the connection. Those cells are then sent along to the destination of the connection. When the equipment at the end of that connection receives the AIS, it sends back FERF cells to the other end of the connection. As shown in Figure 6.15, the AIS and FERF cells specify the type of failure as well as the failure location.

A loopback cell contains a field that specifies whether the cell should be looped back, a correlation tag, a loopback location identification, and a source identification. These loopback cells are used as shown in Figure 6.16.

The device that requests a loopback (we call it the *source*) inserts a loopback cell and selects a value for the correlation tag. The device can specify where the loopback should take place. The device sets the loopback indication field of the cell to 1 to indicate that the cell must be looped back. When the device where the loopback must occur receives the cell, it sets its loopback indication field to 0 and sends the cell back to the source. The source compares the correlation tag of the cell it receives with the value it selected. This correlation tag prevents a device from getting confused by other loopback cells.



6.16

FIGURE

Loopback at the end of connection (above) and at the segment (below).

One other OAM function may be mentioned. Performance management, indicated by OAM cell type 0010, consists of forward monitoring, backward monitoring, and reporting. In forward monitoring, for example, a block of cells of one connection is bounded by OAM cells, which include, among other information, the size of the block. The receiving node can compare the number of received cells with the block size to detect missing or inserted cells.

### 6.5.2 Traffic and Congestion Control

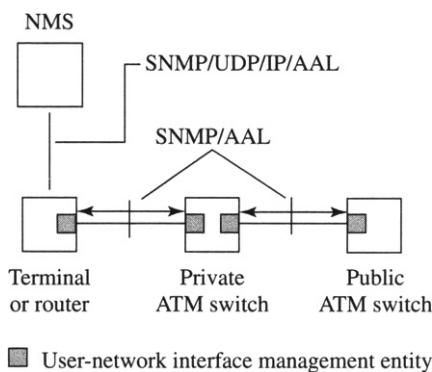
The objectives of traffic and congestion control are to guarantee the contracted quality of service to virtual connections. The operations that the network performs are the subject of Chapters 8 and 9.

### 6.5.3 Network Status Monitoring and Configuration

The OAM functions described above do not provide diagnostic, monitoring, and configuration services across the user-network interfaces. That is the purpose of the *Integrated Local Management Interface* or ILMI protocol, Version 4.0. ILMI uses the Simple Network Management Protocol (SNMP) and a management information base (MIB). The situation is illustrated in Figure 6.17.

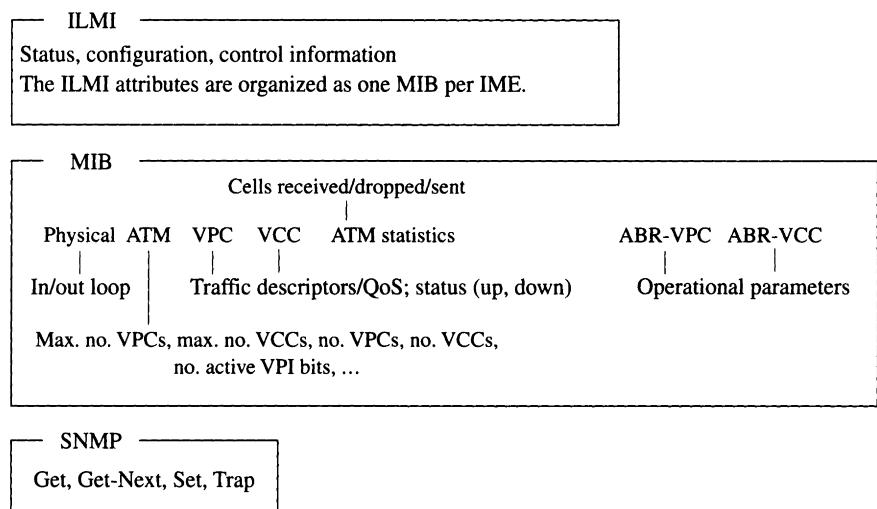
The figure shows a private ATM network connected by a private ATM switch to a public ATM network. Each connection across two interfaces is supervised by two ATM Interface Management Entities (IMEs): one for each of the ATM devices. Two such IMEs are said to be *adjacent*, and the ILMI specifies the structure of the Management Information Base (MIB) that contains the attributes of the connection supervised by the adjacent entities. The ATM devices may be workstations with ATM interfaces that send ATM cells to an ATM switch, or ATM switches, or IP routers that transfer their packets within ATM cells to an ATM switch.

ILMI 4.0 describes four MIB modules. The Textual Conventions MIB defines common textual conventions. The Link Management Module defines the objects for each ATM interface and the methods to detect ILMI connectivity between IMEs. It is further described below. The Address Registration MIB supports procedures for the user and the network to know each other's ATM address. The Service Registry MIB helps to locate ATM network services such as LAN Emulation Configuration Servers (LECS).



6.17

The Integrated Local Management Interface (ILMI) protocol is designed to supervise the connections across user-network interfaces.



**6.18  
FIGURE**

Structure of the ILMI link management MIB that contains the attributes of the connection supervised by adjacent interface management entities.

Important objects of the link management MIB are summarized in Figure 6.18. As the figure indicates, one MIB is defined per IME. The contents of an IME MIB are the attributes of the physical layer (which implements the bit way), the ATM traffic, the VPCs, and VCCs that go across that UNI. The figure indicates representative attributes. The ATM statistics attributes are now deemphasized. The MIB also includes ABR attributes. The ABR virtual path and virtual channel are tuned on a per-connection basis via these attributes. Examples of ABR attributes are ICR (initial cell rate), which is an upper bound on the source's transmission rate; RIF (rate increment factor), which controls the allowed increase in the source transmission rate; and RDF (rate decrement factor), which controls the required decrease in that rate. These attributes are set using ABR resource management (RM) cells, as discussed in section 8.4.2.

### 6.5.4 User/Network Signaling

The basic signaling functions between the network and a user are as follows:

- ◆ the user requests a switched virtual connection,
- ◆ the network indicates whether the request is accepted or not, and
- ◆ the network indicates error conditions with a connection.

We have discussed the UNI signaling protocol above.

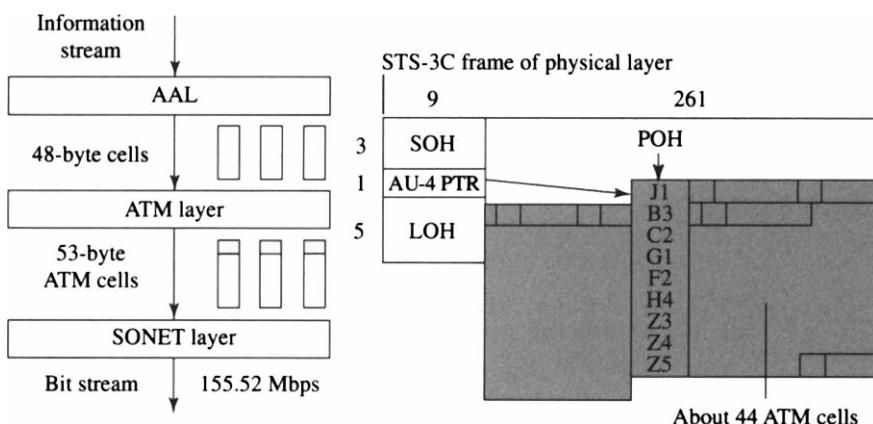
**6.6****BISDN**

The ATM bearer service is the transport of cells with a variety of quality of service. Together with the formats defined for the ATM adaptation layer, this bearer service can be used to support a wide range of applications. In this section we explain how the ATM bearer service in turn can be implemented using SONET networks. The result is an implementation of a Broadband Integrated Services Digital Network, or BISDN.

Figure 6.11 illustrates the BISDN reference model. The standard specifies both transport and physical layers. The network and data link layers are the same as in the ATM standards described above. The physical layer standard has been specified for SONET STS-12C (622.08 Mbps) and STS-3C (155.52 Mbps) signals and for DS3 (44.736 Mbps) signals. We illustrate the ideas for STS-3C. The STS-3C frame, shown in Figure 6.19, is arranged in a  $9 \times 270$  byte matrix.

The first nine columns are devoted to section and line overhead (SOH, LOH). This leaves  $9 \times 261$  bytes, of which one column is devoted to path overhead (POH). The resulting  $9 \times 260$  byte SPE (synchronous payload envelope) contains consecutively arranged 53-byte ATM cells. (Details of SONET can be found in section 5.2.)

The most important features of this frame structure are the following. The ATM bit rate is  $155.52 \times 260/270 = 149.76$  Mbps. The  $9 \times 260 = 2,340$  byte SPE holds about 44.15 53-byte ATM cells. Thus ATM cell boundaries bear no



6.19

BISDN over STS-3C SONET frames.

relationship to the SPE or STS-3C frame boundaries. (The H4 byte in POH is a pointer to the next occurrence of the cell boundary and may be used by the destination to recover cell boundaries. However, the standard now requires recovery of the cell boundary from the header CRC, as explained before.) If the transport layer does not provide a sufficient number of ATM cells, the physical layer inserts idle ATM cells into the frame, which are removed by the physical layer at the destination. As with SONET traffic generally, the SPE is not aligned with the STS-3C frame. The SPE location is obtained from the AU-4 pointer in the LOH.

Future subscribers to BISDN might receive an STS-12 (622.08-Mbps) signal, and they will be able to send an STS-3C signal. The subscriber can share this bandwidth among many simultaneous connections of varying QoS: constant and variable bit rate video and audio traffic with real-time constraints, data traffic with guaranteed retransmission in case of error for file transfer traffic, datagrams traffic for short transactions such as database queries or remote procedure calls, and so on. The ATM/BISDN standard is sufficiently flexible to accommodate this variety of traffic service.

BISDN is a product of a telephone-centric view of networking. The standards pay much attention to interconnecting public telephone carriers, and they assumed that ATM was flexible enough to support all services. Over time, however, the situation seems to have reversed, and ATM has taken over the burden of internetworking. BISDN has been relegated into the background.

## **6.7 INTERNETWORKING WITH ATM**

An ATM network can be used to carry internetwork traffic. For instance, an ATM network can be used to interconnect various LANs or IP subnetworks. Such internetworking can take place at the data link layer (bridging) or at the network layer (routing). For more details on the material of this section, we refer the reader to [AL95].

Internetworking is a major challenge facing ATM. Two basic tasks are required for internetworking over ATM. The first is encapsulation of the protocol data units, and the second is the routing or bridging of these PDUs. The routing itself consists of the route calculation and the switching of packets. The route calculation necessitates the address resolution plus a routing algorithm. The address resolution maps the protocol address (such as IP, MAC, FR, or SMDS) into an ATM address. The routing algorithm calculates the routes through the

network. That routing algorithm for the private ATM network is PNNI, as we learned in section 6.2.2. The non-ATM network can use either its own routing algorithm or an extension of PNNI.

In the following subsections we explain encapsulation, LAN emulation, IP over ATM, a more general multiprotocol over ATM, and FR or SMDS over ATM.

It should be noted that internetworking involves other tasks not considered here. These are signaling interworking and user data transfer internetworking between different protocols.

### 6.7.1 Multiprotocol Encapsulation over AAL5

The encapsulation of internetwork traffic is defined in RFC 1483. The RFC specifies two methods depending on whether one ATM VCC carries a single or multiple protocols. In either case, the protocol data units are carried by the payload of the convergence sublayer of AAL5. That is, a PDU of up to 64 KB is first padded so that, together with the 8-byte convergence sublayer trailer that specifies the length of the PDU and a CRC, it adds up to a multiple of 48 bytes.

If the VCC carries a single protocol, this protocol is identified implicitly by the VPI/VCI. Otherwise, the convergence sublayer PDU starts with a header that specifies the protocol (e.g., routed IP or bridged 802.3-6).

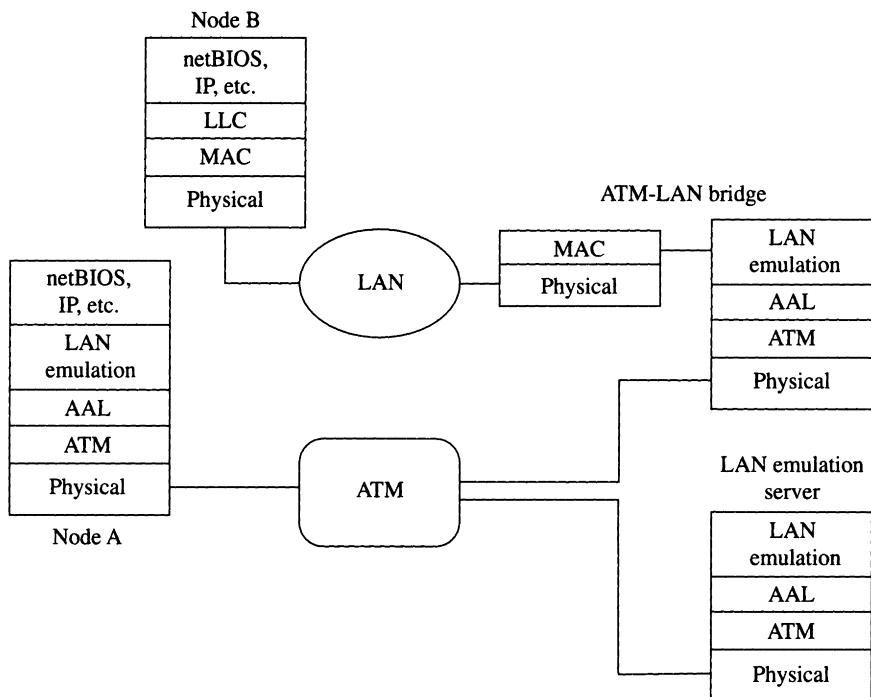
If the protocol is bridged, then the MAC address of the destination must be specified in the convergence sublayer PDU. Note that the ATM interface must perform the usual functions of a bridge with dynamic learning by looking into the MAC addresses of the encapsulated PDUs.

We explain how such a bridging function is implemented in LAN interconnections in the next section.

### 6.7.2 LAN Emulation over ATM

LAN emulation (LANE) is a glue that enables ordinary LAN software to operate over ATM and also to interconnect LANs and ATM. (See [A95b].) This emulation enables the connection of Ethernets through an ATM network or of token rings through an ATM network (not a mixture).

Protocols, such as IP's Address Resolution Protocol (ARP), that are dependent on the availability of a broadcast function, are supported by LANE over ATM, which, due to its connection-oriented nature, is nonbroadcast. ARP uses broadcast to resolve internetwork layer addresses to MAC addresses. On an Emulated LAN (ELAN), LANE supports this broadcast with a Broadcast/Unknown



6.20  
FIGURE

A LAN emulation layer is inserted between the network layer and the AAL layer in ATM nodes.

Server (BUS). However, to effect the actual data transfer over an ATM network, a further mapping from MAC address to ATM address is necessary. The LANE server provides this mapping. The host then transfers the data by setting up an ATM VCC to the target ATM address.

Figure 6.20 illustrates this approach. A packet destined for B invokes the LAN emulation layer that contacts the server to find the ATM address of the bridge. The packet is segmented and sent to the bridge, whose LAN emulation software reconstructs the packet before sending it on the LAN. The reverse transfer, from B to A, is similar.

Broadcast packets and packets with unknown destination addresses are flooded by the server.

This solution enables ATM networks to interoperate with existing "legacy" LANs.

### 6.7.3

### IP over ATM

IP is a powerful and widely used internetworking protocol. With IP we can interconnect IEEE 802 networks easily. IP is a datagram network layer. In this chapter we discussed the ATM technology and how it can be used to build local area networks and wide area networks with a good control on the quality of service it provides to applications.

For the ATM technology to be widely implemented, it must interoperate with the IP protocol suite. In this section we explain how ATM networks can transport IP packets. This possibility enables a progressive upgrade of the Internet to the ATM technology. The advantage such an upgrade would provide is that applications requiring the tight control of QoS can be supported by ATM and not easily by the TCP/IP protocols. Thus, progressively, the Internet would evolve into a BISDN network while remaining compatible with the installed base of best-effort services.

We explain three strategies: the classical IP model, the shortcut models, and the integrated models. We then explain multicast IP over ATM. These strategies are being evolved by the IETF working group IP over ATM. (See RFC 1754.)

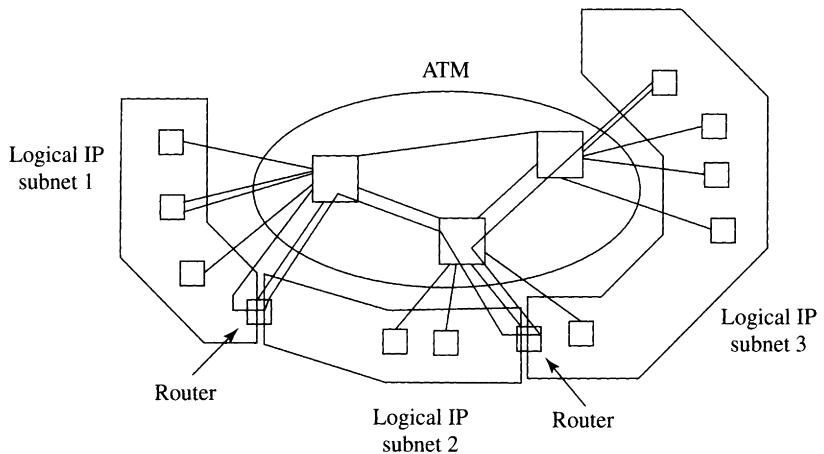
#### *Classical IP*

The use of LANE for IP transfer requires two levels of address resolution: the IP address must be resolved to a MAC address, and the MAC address must be resolved to an ATM address using LANE. Classical IP directly provides IP-to-ATM address resolution using an ARP server, thereby reducing broadcast traffic.

Consider the situation shown in Figure 6.21. In the classical IP model, the nodes attached to an ATM network are grouped into logical IP subdomains (LIS). Routing between logical IP subdomains is via routers, as shown in the figure. Note that AAL5 is used so that the router reassembles the packet before forwarding.

Within one given logical IP subdomain, a node uses an address-resolution protocol (ARP) server. The stations all know the ATM address of their ARP server. Thus, to find a particular destination, instead of broadcasting a request *are you node IP.address?* to find the physical address, here a node sends a request to the ARP server of the subdomain asking, *what is the VCI of a particular IP.address?*

In the case of SVCs (switched virtual circuits), the nodes need to register with the ARP server. They do so by calling the ARP server (using the ATM



**6.21**  
**FIGURE**

Nodes attached to an ATM network are grouped into logical IP subdomains (LIS). The routing is via routers between subdomains. Within one domain, the routing uses ATM ARP with an ARP server.

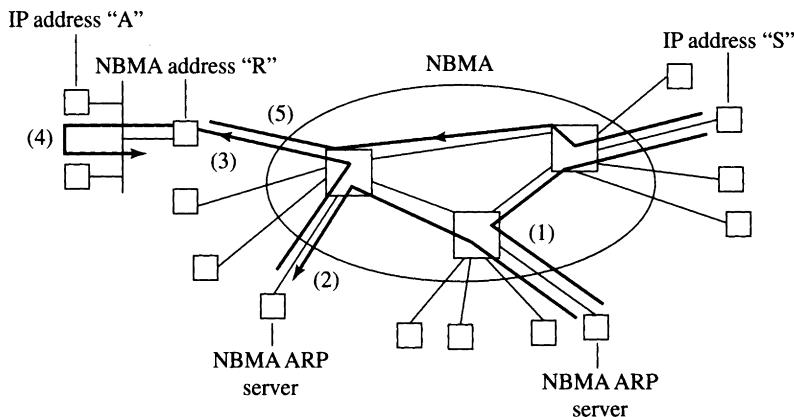
addresses). The server then asks, *what is your IP address?* and enters that information in its table [IP → ATM].

The IP and ARP packets are encapsulated over AAL5. Two alternatives exist: either one VC is allocated per protocol (one for IP, one for ARP), or multiple protocols are multiplexed over a single VC per subnetwork attachment point (SNAP). The maximum transmission unit in IP over ATM is fixed to 9,180 bytes. Other sizes (up to 64 KB) can be agreed on by configuration (in the case of PVC or permanent virtual circuit) or by signaling (for SVC). The type of ATM connection is either CBR or VBR with specified peak rates forward and backward. Other encapsulations are being proposed to eliminate or reduce the redundant IP header (See RFC 1483, 1755, 2225.)

### ***Shortcut Models***

Instead of retransmitting via routers as in the classical IP model, the idea of the shortcut models is to go directly from source to destination ATM nodes. In the accepted terminology, the ATM network is called a *nonbroadcast multiaccess* (NBMA) link layer. In Figure 6.22 we indicate how a node S finds the NBMA address.

The NBMA Next Hop Resolution Protocol (NHRP) extends the idea above. It allows a source wishing to communicate over a NBMA subnetwork to de-



6.22

**FIGURE**

To transmit to node A, node S finds out the nonbroadcast multiaccess (NBMA) link address of the router R according to the following steps: (1) What is the NBMA address of A? (2) forwards request; (3) knows router of network of A; (4) ARP to locate A; then, in reverse, (3), (2), (1) gives NBMA address R; a connection is then made as shown in (5).

termine the internetworking layer addresses and NBMA addresses of suitable “NBMA next hops” toward a destination station. Routers are required to interconnect these subnets, but NHRP allows intermediate routers to be bypassed on the data path. NHRP provides an extended address resolution protocol, which permits Next Hop Clients to send queries between different subnets. Queries are propagated by Next Hop Servers along the routed path determined by a standard routing protocol. This enables the establishment of ATM VCCs across subnet boundaries without routers in the data path.

Note that this method can also be used for Frame Relay, ISDN, and X.25 networks that do not support broadcasting. (See RFC 1735, 2332.)

### ***Integrated Model***

The integrated model proposal aims to simplify the routing by integrating addressing and routing of IP and ATM. In this model, the ATM address could be a superset of the IP address. (This approach does not work with networks that use the E.164 addresses.) The IP router then maps the destination IP address into the ATM address of the destination if it is directly reachable or of the best router otherwise. The selection of the best router may be load dependent.

### Multicast IP over ATM

The main difference between ATM multicast VCs and multicast IP is that the source must add new destinations in the case of ATM.

The mechanism uses a Multicast Address Resolution Server (MARS) that maps an IP multicast address to either the list of all the individual ATM addresses or to the ATM address of a Multicast Server (MCS). A *cluster* is a set of hosts that use the same MARS. The communication between clusters works as the regular IP multicast routing. We describe the communication within one cluster.

There are two approaches to multicast connections within one cluster. In the direct approach, each sender sets up one VC per member of the multicast group. In the indirect approach, the sender sends to an MCS, which then sets up VCs to the group members.

In the direct approach, the hosts send join and leave requests to the MARS. The MARS maintains a point-to-multipoint VC to all the sources (called the ClusterControlVC) to inform them of group changes. The join and leave messages are retransmitted over the ClusterControlVC. Before sending, the source asks MARS for a list of ATM addresses. Then the sender sets up the VCs.

In the indirect approach, the sender knows only the MCS as a member of a group. The hosts register with MCS, and every MCS registers with MARS. (See RFC 2022.)

### 6.7.4 Multiprotocol over ATM (MPOA)

To increase the penetration of ATM technology, the ATM Forum issued MPOA version 1.0. The objective of MPOA is to support the efficient transfer of inter-subnet unicast traffic in a LANE environment. MPOA integrates LANE and NHRP, allowing the transfer of layer 3 packets (typically IP) over fast ATM VCCs without requiring routers in the data path. MPOA allows the physical separation of the internetwork (IP) layer route calculation and the forwarding of the data. The technique is called *virtual routing*. Virtual routing can decrease the number of devices participating in internetwork layer route calculations, increasing scalability.

MPOA provides MPOA clients (MPC) and MPOA servers (MPS) and defines the protocols used by MPCs to issue queries for shortcut ATM addresses and receive replies from MPSs. The shortcut is an ATM VCC from an ingress MPC source to an egress MPC sink through an ATM cloud (MPOA system). Until a

shortcut is obtained, a default routed path is used. The technique is similar to tag switching in which layer 3 routing table lookups are bypassed.

The main function of the MPC is to serve as a shortcut source and sink. The MPC performs forwarding but does not run routing protocols. In its ingress role, an MPC detects packet flows that are being forwarded over an emulated LAN to a router that contains an MPS. When the MPC recognizes a flow (for example, by counting the number of packets in the flow over a certain duration) that could benefit from a shortcut that bypasses the default route, it uses an NHRP-based query to request a shortcut to the destination. If the shortcut is available, the MPC caches this information, sets up a shortcut VCC, and forwards the packets over the shortcut. The shortcut information may be in the form of a 32-bit tag that the egress MPC provides to the ingress MPC. The tag is included by the ingress MPC in the MPOA packet header.

In its egress role the MPC receives the packets from other MPCs to be forwarded to its local users. For frames received over a shortcut, the egress MPC adds the appropriate data link layer (MAC) encapsulation.

The MPOA server is the logical part of a router that provides forwarding information to MPCs using NHRP.

### 6.7.5 FR and SMDS over ATM

The transport of Frame Relay packets over ATM requires the conversion of the FR data link connection identifier into a VPI/VCI for the ATM connection and the segmentation of the FR packet payload into AAL5 packets.

The FR congestion and discard eligibility fields are mapped into the ATM EFCI bit and the ATM CLP bit, respectively. Moreover, the FR committed information rate is mapped into VRB traffic parameters.

SMDS packets are mapped into AAL3/4 cells and transported over a well-known VPI/VCI. A connectionless server within the ATM network receives the cells and forwards the packet.

## 6.8

## SUMMARY

In the two previous chapters we studied packet- and circuit-switched networks. Packet-switched networks handle well message traffic that imposes very loose delay constraints. Circuit-switched networks are well suited for constant bit rate traffic with hard delay constraints. Of course, both networks can accommodate all types of traffic, but they may do this inefficiently.

Engineers invented ATM networks with the objective that they would provide a highly flexible bearer service capable of supporting in an efficient manner applications that range from those that need guaranteed delay and loss bounds to those that need only best-effort service provided by datagram networks such as Internet. At the end of the 1980s, ATM networks were exotic topics of discussion at research conferences. Today, many vendors provide ATM equipment, although with limited functionality. Some ATM switches could already be classified as carrier grade and satisfy requirements in terms of speed (2.5 Gbps ports, 50 Gbps backplane), reliability (PNNI rerouting) and overall functionality.

This chapter was devoted to a discussion of the functionality of ATM. In principle ATM can offer the full range of services needed to make the claim that ATM networks can function as the "universal network." The great interest among telephone companies in providing ATM over SONET suggests that ATM will be a serious contender for this title, in the name of BISDN. However, BISDN is still a considerable distance in the future. The more immediate goal is the migration path for IP over ATM. The migration can permit an "upgrade" of IP in the sense that IP/ATM may be able to provide service guarantees that IP cannot. If this turns out to be the case, IP and ATM may gain through economies of scope and service integration.

Tremendous progress in ATM has been made over the 1990s. But the potential of ATM will remain unfulfilled until the difficult issues of resource allocation and control are satisfactorily resolved such that the same ATM network can efficiently provide a variety of service qualities. In Chapters 8 and 9 we discuss these issues. At the present time, however, the challenge facing ATM is attracting IP.

## 6.9

## NOTES

Specifications of the ATM Forum are now available at [www.atmforum.com](http://www.atmforum.com). The parameters and procedures related to traffic management and QoS are specified in [A96d]. GFR is likely to appear in ATM Forum's Traffic Management Specification version 4.1. GFR is discussed in [GJF98]. ATM addressing is specified in [A99]. The basic ATM formats and protocols are discussed in [B99, P94]. FUNI is specified in [A97a]. The BISDN model is described in [A95a], which also specifies interworking with FR and SMDS.

The all-important management functions that guarantee the service quality level, however, are not yet sufficiently defined to constitute a proposal. Nevertheless, there is important and continuing discussion within the ATM

Forum that seeks to develop those proposals and standards. PNNI 1.0, UNI 4.0, ILMI 4.0 are specified in [A96a, A96b, A96c].

LAN emulation is specified in [A95b], and MPOA in [A97b]. Proposals for IP over ATM are discussed in various IETF RFC, notably RFC 1483, 1755, 2225, 2332. MPOA is compared with Multiprotocol Label Switching (MPLS) in [DDR98].

## 6.10

## PROBLEMS

- Suppose the transport layer transfers fixed-size packets of  $N$  bits either as datagrams or over a virtual circuit. In the former case  $n_d$  bits are used to encode the destination address. In the latter case  $n_c < n_d$  bits are used to encode the VCI. In addition, in the latter case, there is a fixed delay incurred to set up the connection. We measure this delay in terms of the time needed to transmit  $D$  bits. Suppose the source wants to transmit a message of  $M$  bits. Will the datagram or the connection-oriented service incur greater delay?
- How many sequence numbers are needed to detect cell loss? How many are used in SMDS?
- The packetization delay depends on the speed of the information transfer. Calculate the packetization delay for (a) 53-byte ATM cells and (b) a 1,000-byte packet transfer service for (1) voice samples that are sampled 8,000 times per second and encoded into a 64-Kbps stream and (2) MPEG1, which takes 30 video frames per second and encodes them into a 1-Mbps stream.
- The size of the ATM cell affects cell error rate. If the transmission system has a bit error rate (BER) of  $p$ , and there are  $N$  bits per cell, show that the cell error rate (CER) is

$$CER = 1 - (1 - p)^N \approx Np.$$

Suppose that a cell is retransmitted whenever it contains an error. Then the average number of cells transmitted per error-free cell reception is  $(1 - CER)^{-1}$ . Suppose the overhead per cell is fixed at  $n$  bits, independent of cell size. Show that the average number of bits transmitted per error-free reception of one bit of information is

$$\frac{N + n}{N(1 - p)^{N+n}}.$$

For a given BER,  $p$ , and overhead  $n$ , the optimal cell size  $N$  minimizes this number. Taking  $n = 40$  (5-byte overhead), find the best  $N$  for two extreme cases:  $p = 10^{-9}$  and  $p = 10^{-3}$ .

5. How many simultaneous connections are needed to support all foreseeable needs, including telephone, video, and text? Will the 16-bit VCI be enough?
6. Compare the number of bits devoted for IP addressing with the requirements for ATM. Suppose you want to transmit voice in small IP packets so that the packetization delay is not more than  $X$  ms. How large a packet can you tolerate, and what is the overhead rate (ratio of number of overhead bits to packet size)?
7. Suppose voice is sampled and transported over ATM. Suppose the cell stream is subject to random delay. How would you characterize the resulting distortion?
8. Using LAN emulation, hosts attached to Ethernet switches with an ATM backbone
  - (a) get a guaranteed quality of service.
  - (b) cannot select the parameters of the virtual circuits.
  - (c) may benefit from a well-provisioned backbone.
  - (d) must implement AAL-5.
9. Please answer the following questions in one or two sentences.
  - (a) Why is “out-of-band signaling” preferable to “in-band signaling”?
  - (b) Why do ATM networks use cell switching?
  - (c) Why is virtual circuit switching critical for ATM networks? (Please list three reasons.)
  - (d) What are the “adaptation layers”? Are they in the switches? Are they in the end-hosts?
  - (e) What are the relative advantages and disadvantages of using ATM in the “native mode” vs. “LAN Emulation”?
  - (f) Is it practical to build a large (i.e., with lots of ports) *input-buffered* ATM switch with fast links? Why or why not?
  - (g) Is it practical to build a large (i.e., with lots of ports) *output-buffered* ATM switch with fast links? Why or why not?
  - (h) Assume you have a choice between an input-buffered and an output-buffered ATM switch, each with the same number of ports and same speed links, which is better? Explain why.

# Wireless Networks

**F**uture wireless networks will enable people on the move to communicate with anyone, anywhere, at any time, using a range of multimedia services. The exponential growth of cellular telephone and paging systems coupled with the proliferation of laptop and palmtop computers indicate a bright future for such networks, both as standalone systems and as part of the larger networking infrastructure.

This chapter describes wireless networks. Section 7.1 provides an introduction to these networks, including their history and prospects, and the technical challenges of design and operation posed by the underlying wireless channel. Section 7.2 presents the main characteristics of the wireless channel and their impact on the link and network layer design. Link layer design techniques developed to overcome wireless channel impairments to delivering high data rates with low distortion are described in Section 7.3.

The wireless channel is a limited resource that is shared among many users. Section 7.4 is devoted to channel access protocols. Section 7.5 outlines design issues for wireless networks, including network architecture, user location and routing protocols, network reliability and QoS, internetworking between wireless and wired networks, and security.

Current wireless network technology is described in Section 7.6 including cellular and cordless telephones, wireless LANs and wide area data services, paging systems, and global satellite systems. Section 7.7 gives an overview of emerging systems and standards for future wireless networks. Section 7.8 contains a summary and a discussion of future trends.

## 7.1 INTRODUCTION

Wireless communications is the fastest growing segment of the communications industry. Cellular phones, cordless phones, and paging services have experienced exponential growth over the last decade, and this growth continues unabated worldwide. Wireless communications has become a critical business tool and part of everyday life in most developed countries. Wireless communication systems are replacing antiquated wireline systems in many developing countries. Will future wireless networks live up to their promise of multimedia communications anywhere and any time? In this introduction we review the history of wireless networks. We then discuss the wireless vision in more detail, including the technical challenges that must be overcome to make this vision a reality.

### 7.1.1 History of Wireless Networks

The first wireless networks were developed in the preindustrial age. These systems transmitted information over line-of-sight distances (later extended by telescopes) using smoke signals, torch signaling, flashing mirrors, signal flares, and semaphore flags. An elaborate set of signal combinations was developed to convey complex messages with these rudimentary signals. Observation stations were built on hilltops and along roads to relay these messages over large distances. These early communication networks were replaced first by the telegraph network (invented by Samuel Morse in 1838) and later by the telephone. In 1895, twenty years after the telephone was invented, Marconi demonstrated the first radio transmission from the Isle of Wight to a tugboat 18 miles away, and radio communications was born. Radio technology advanced rapidly to enable transmissions over larger distances with better quality, less power, and smaller, cheaper devices, thereby enabling public and private radio communications, television, and wireless networking.

Early radio systems transmitted analog signals. Today most radio systems transmit digital signals composed of binary bits, where the bits are obtained directly from a data signal or by digitizing an analog voice or music signal. A digital radio can transmit a continuous bit stream or it can group the bits into packets. The latter type of radio is called a *packet radio* and is characterized by bursty transmissions: the radio is idle except when it transmits a packet. The first packet radio network, Alohanet, was developed at the University of Hawaii in 1971. This network enabled computer sites at seven campuses spread

out over four islands to communicate with a central computer on Oahu via radio transmission. The network architecture used a star topology with the central computer at its hub. Any two computers could establish a bidirectional communications link between them by going through the central hub.

Alohanet incorporated the first set of protocols for channel access and routing in packet radio systems, and principles underlying these protocols are still in use today. Activity in packet radio, promoted by DARPA, peaked in the mid 1980s, but the resulting networks fell far short of expectations in terms of speed and performance. Packet radio networks today are mostly used by commercial providers of wide area wireless data services. These services, first introduced in the early 1990s, enable wireless data access (including e-mail, file transfer, and Web browsing) at fairly low speeds, on the order of 20 Kbps. The market for these data services has not grown significantly.

In the 1970s Ethernet technology steered companies away from radio-based networking. Ethernet's 10 Mbps data rate far exceeded anything available using radio. In 1985 the Federal Communications Commission (FCC) enabled the commercial development of wireless LANs by authorizing the public use of the Industrial, Scientific, and Medical (ISM) frequency bands for wireless LAN products. The ISM band was very attractive to wireless LAN vendors since they did not need to obtain an FCC license to operate in this band. However, the wireless LAN systems could not interfere with the primary ISM band users, which forced them to use a low power profile and an inefficient signaling scheme. Moreover, the interference from primary users within this frequency band was quite high. As a result these initial LAN systems had very poor performance in terms of data rates and coverage.

The poor performance, coupled with concerns about security, lack of standardization, and high cost (the first network adaptors listed for \$1,400 as compared with a few hundred dollars for a wired Ethernet card) resulted in weak sales for the initial wireless LAN systems. The current generation of wireless LANs, based on the IEEE 802.11 standard, has better performance, although the data rates are still low (on the order of 2 Mbps) and the coverage area is still small (around 500 feet). Ethernets today offer data rates of 100 Mbps, and the performance gap between wired and wireless LANs is likely to increase over time without additional spectrum allocation. Thus, it is not clear if wireless LANs will be competitive except where users sacrifice performance for mobility or when a wired infrastructure is not available.

By far the most successful application of wireless networking has been the cellular telephone system. Cellular telephones have approximately 200 million subscribers worldwide, and their growth continues at an exponential pace. The convergence of radio and telephony began in 1915, when wireless

voice transmission between New York and San Francisco was first established. In 1946 public mobile telephone service was introduced in 25 cities across the United States. These initial systems used a central transmitter to cover an entire metropolitan area. This inefficient use of the radio spectrum coupled with the state of radio technology at that time severely limited the system capacity: thirty years after the introduction of mobile telephone service the New York system could only support 543 users. A solution to this capacity problem emerged during the fifties and sixties when researchers at AT&T Bell Laboratories developed the cellular concept.

Cellular systems exploit the fact that the power of a transmitted signal falls off with distance, so the same frequency channel can be allocated to users at spatially separate locations with minimal interference. A cellular system divides a geographical area into adjacent, nonoverlapping "cells." Cells assigned the same channel set are spaced apart so that interference between them is small. Each cell has a centralized transmitter and receiver (called a *base station*) that communicates with the mobile units in that cell, both for control purposes and as a call relay. All base stations have high-bandwidth connections to a mobile telephone switching office (MTSO), which is itself connected to the public-switched telephone network (PSTN). The handoff of mobile units crossing cell boundaries is typically handled by the MTSO, although in current systems some of this functionality is handled by the base stations and/or mobile units.

The original cellular system design was finalized in the late 1960s and deployed in the early 1980s. The large and unexpected growth led to the development of digital cellular technology to increase capacity and improve performance.

The current generation of cellular systems are all digital. In addition to voice communication, these systems provide e-mail, voice mail, and paging services. Unfortunately, the great market potential for cellular phones led to a proliferation of digital cellular standards. Today there are three different digital cellular phone standards in the U.S. alone, and other standards in Europe and Japan, none of which are compatible. The incompatible standards make roaming throughout the U.S. using one digital cellular phone impossible. Most cellular phones today are dual-mode: they incorporate one of the digital standards along with the old analog standard that provides coverage throughout the U.S.

Radio paging systems represent another example of a successful wireless data network, with 50 million subscribers in the U.S. Their popularity is starting to wane with the widespread penetration and competitive cost of cellular telephone systems. Paging systems allow coverage over very wide areas by

simultaneously broadcasting the pager message at high power from multiple base stations or satellites. Early radio paging systems were analog 1-bit messages signaling a user that someone was trying to reach him or her. These systems required callback over the regular telephone system to obtain the phone number of the paging party.

Paging systems now allow a short digital message, including a phone number and brief text, to be sent to the pagee. In paging systems most of the complexity is built into the transmitters, so that pager receivers are small, lightweight, and have a long battery life. The network protocols are also very simple since broadcasting a message over all base stations requires no routing or handoff. The spectral inefficiency of these simultaneous broadcasts is compensated by limiting each message to be very short. Paging systems continue to evolve to expand their capabilities beyond very low-rate one-way communication. Current systems are attempting to implement two-way, "answer-back" capability. This requires a major change in pager design, since it must now transmit signals in addition to receiving them, and the transmission distances can be quite large.

Commercial satellite communication systems form another major component of the wireless communications infrastructure. They provide broadcast services over very wide areas and help fill the coverage gap between high-density user locations. Satellite mobile communication systems follow the same basic principle as cellular systems, except that the cell base stations are now satellites orbiting the earth. Satellite systems are typically characterized by the height of the satellite orbit, low-earth orbit (LEO), medium-earth orbit (MEO), or geosynchronous orbit (GEO).

The idea of using geosynchronous satellites for communications was first suggested by the science fiction writer Arthur C. Clarke in 1945. However, the first deployed satellites, the Soviet Union's *Sputnik* in 1957 and the NASA-Bell Laboratories *Echo-1* in 1960, were not geosynchronous due to the difficulty of lifting a satellite into such a high orbit. The first GEO satellite was launched by Hughes and NASA in 1963, and from then until very recently GEOs dominated both commercial and government satellite systems. The current trend is to use lower orbits so that lightweight handheld devices can communicate with the satellite. Services provided by satellite systems include voice, paging, and messaging services, all at fairly low data rates.

## 7.1.2 Wireless Data Vision

Wireless communication providing high-speed, high-quality information exchange between portable devices located anywhere in the world is the vision

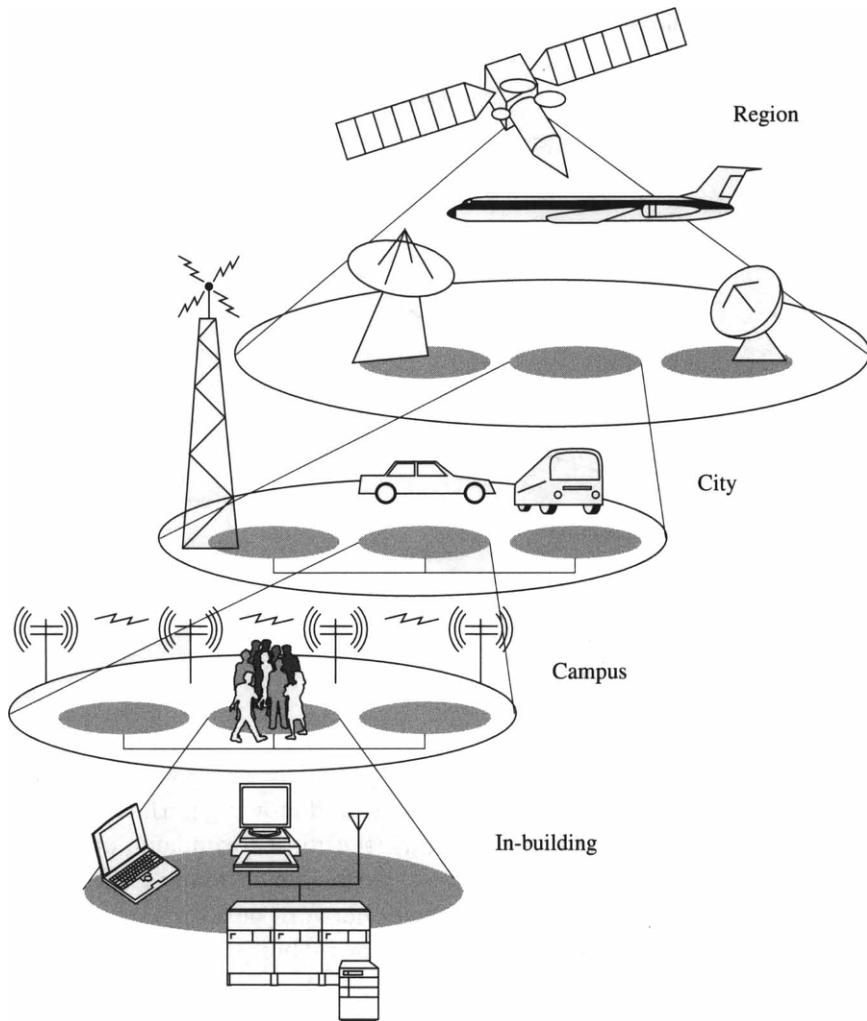
for the next century. People will operate a virtual office anywhere in the world using a small handheld device—with seamless telephone, modem, fax, and computer communications. Wireless LANs will connect together palmtop, laptop, and desktop computers anywhere within an office building or campus, as well as from the corner cafe. In the home these LANs will enable intelligent home appliances to interact with one another and with the Internet. Video teleconferencing will take place between buildings that are blocks or continents apart, and these conferences may include travelers. Wireless video will be used to create remote classrooms, remote training facilities, and remote hospitals anywhere in the world.

Wireless communications may be classified in terms of applications, systems, or coverage regions, as indicated in Figure 7.1. The applications include voice, Internet access, Web browsing, paging and short messaging, subscriber information services, file transfer, and video teleconferencing. Systems include cellular telephone systems, wireless LANs, wide area wireless data systems, and satellite systems. Coverage regions include in-building, campus, city, regional, and global.

Wireless applications have different requirements, as we saw in Chapter 2. Voice systems have low data rate requirements (around 20 Kbps) and can tolerate a high bit error rate (BER) of  $10^{-3}$ , but the total delay must be less than 100 ms. Data systems may require higher data rates (1–100 Mbps) and very small BER, but do not have a fixed delay requirement. Real-time video systems have high data rate requirements coupled with the same delay constraints as voice systems, while paging and short messaging have very low data rate requirements and no delay constraints.

These diverse requirements make it difficult to build a single wireless system that can satisfy them all. As we have seen, IP and ATM can support diverse requirements on wired networks, with data rates on the order of Gbps and BERs on the order of  $10^{-12}$ . But this is not possible on wireless networks, which have much lower data rates and higher BERs. Therefore wireless systems will continue to be fragmented, with different protocols tailored to support different requirements.

It is uncertain whether there will be a large demand for all wireless applications. Companies are investing heavily to build multimedia wireless systems, yet the only highly profitable wireless application so far is voice. Despite optimistic predictions, the market for these products remains relatively small with sluggish growth. To examine the future of wireless data, it is useful to see the growth of various communication services over the last five years, as shown in Figure 7.2. The exponential growth of cellular and paging subscribers is exceeded only by the demand for Internet access. The number of laptop and



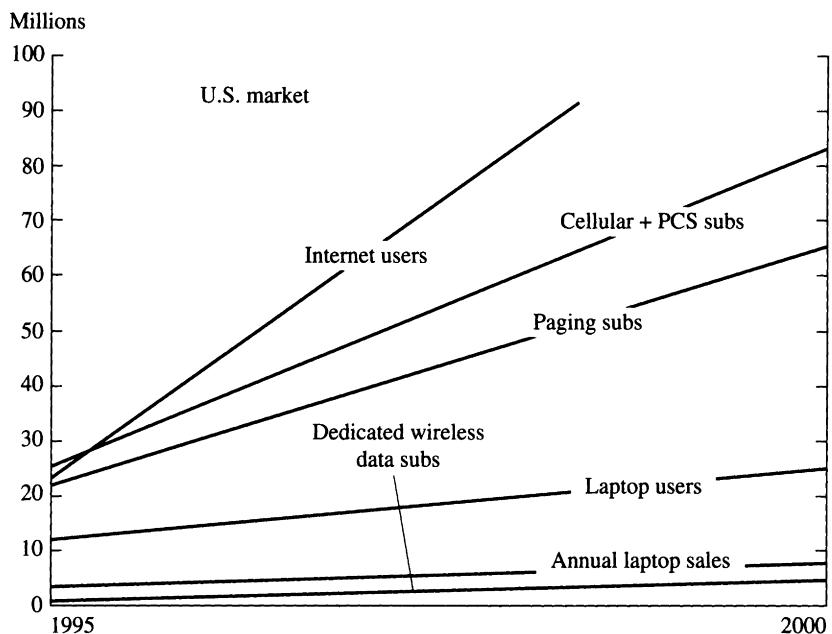
---

7.1

Wireless applications, systems, and coverage regions.

**FIGURE**

palmtop computers is also growing steadily. These trends indicate that people want to communicate while on the move. They also want to take their computers wherever they go. It is therefore reasonable to assume that people want the same data communications capabilities on the move as they enjoy in their home or office. Yet the number of wireless data subscribers remains relatively flat, most likely because of the high cost and poor performance of today's systems.



7.2

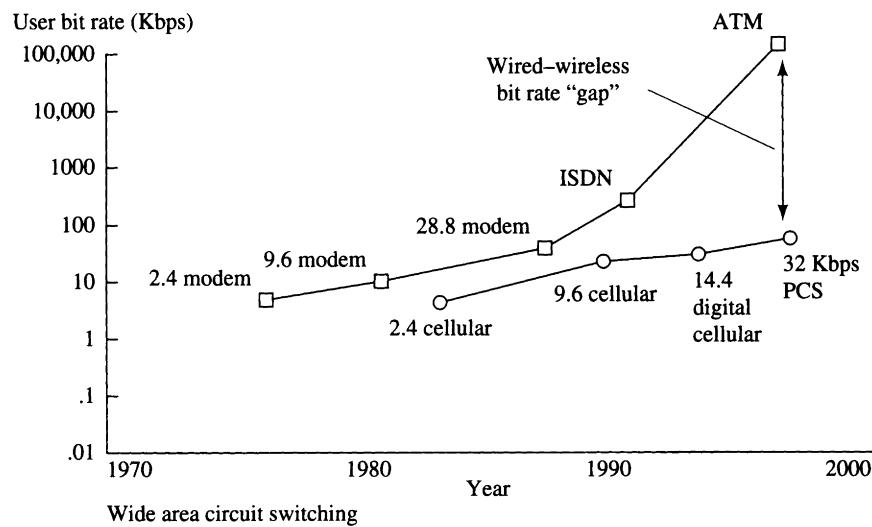
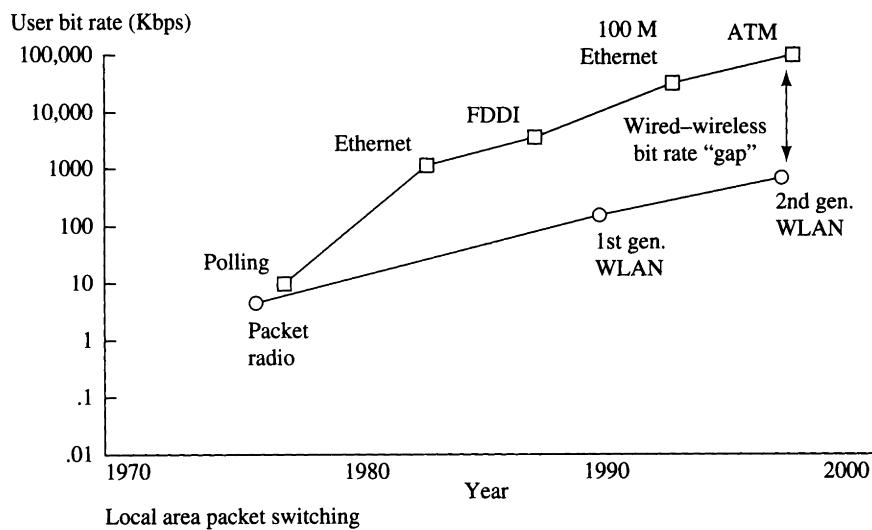
FIGURE

Growth of wireless communication services.

Figure 7.3 displays the large and growing performance gap between wired and wireless networks. Thus, the most formidable obstacle to the growth of wireless data systems is their performance. Many technical challenges must be overcome to improve wireless network performance so that users will accept this performance in exchange for mobility.

### 7.1.3 Technical Challenges

Technical problems must be solved across all levels of the system design to realize the wireless vision. At the hardware level the terminal must have multiple modes of operation to support the different applications and different media. Desktop computers process voice, image, text, and video data, but breakthroughs in circuit design are required to implement multimode operation in a small, lightweight, handheld device. Since most people don't want to carry a heavy battery, the signal processing and communications hardware of the portable terminal must consume very little power, which will impact higher levels of the system design.



Many signal processing techniques for efficient spectral utilization and networking demand much processing power, precluding the use of low-power devices. Hardware advances for low-power circuits with high processing ability will relieve some of these limitations. However, placing the processing burden on fixed location sites with large power resources will as in the past continue to dominate wireless system designs. But the associated bottlenecks and single points-of-failure are undesirable. The finite bandwidth and random variations of the communication channel will also require robust compression schemes that degrade gracefully with the channel.

The wireless channel is an unpredictable and difficult communications medium. The scarce radio spectrum is regulated. In the U.S. spectrum is allocated by the FCC. In Europe the equivalent body is the European Telecommunications Standards Institute (ETSI), and globally spectrum is controlled by the International Telecommunications Union (ITU). A regional or global system operating in a given frequency band must obey the regulatory restrictions.

Spectrum has become expensive, especially in the U.S., since the FCC began auctioning spectral allocations. In the recent spectral auctions at 2 GHz, companies spent over \$9 billion for licenses. The spectrum obtained through these auctions must be used extremely efficiently to get a reasonable return on investment, and it must also be reused over and over in the same geographical area, thus requiring cellular system designs with high capacity and good performance. At frequencies around several GHz, wireless radio components with reasonable size, power consumption, and cost are available. However, the spectrum in this frequency range is extremely crowded. Technological breakthroughs that enable higher frequency systems with the same cost and performance would greatly reduce the spectrum shortage, although path loss at these higher frequencies increases, thereby limiting range.

As a signal propagates through a wireless channel, it experiences random fluctuations in time if the transmitter or receiver is moving, due to changing reflections and attenuation. This makes it difficult to design reliable systems with guaranteed performance. Security is also more difficult to implement in wireless systems, since the airwaves are susceptible to snooping from anyone with an RF antenna. Analog cellular systems have no security, and you can easily listen in on conversations by scanning the analog cellular frequency band. To support applications like electronic commerce and credit card transactions, the wireless network must be secure against such listeners.

Wireless networking presents another set of problems. The network must locate a user among millions of globally distributed mobile terminals. It must then route a call to that user moving at speeds of up to 100 mph. The finite resources of the network must be allocated in a fair and efficient manner

relative to changing user demands and locations. Different problems are posed by the need for protocols to interface between wireless and wired networks with vastly different performance capabilities.

Perhaps the most significant technical challenge in wireless network design is an overhaul of the design process itself. Wired networks are mostly designed according to the layers of the OSI model: each layer is designed independently from the other layers with mechanisms to interface between layers. This greatly simplifies network design, although it leads to some inefficiency and performance loss. The situation is very different in a wireless network. Wireless link performance, user connectivity, and network topology change over time. To achieve good performance wireless networks must adapt to these changes, requiring an integrated and adaptive protocol stack across all layers, from the link layer to the application layer.

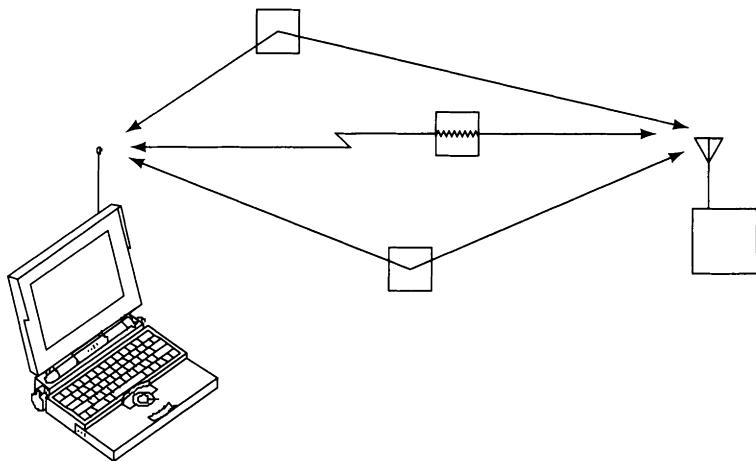
## 7.2

## THE WIRELESS CHANNEL

The wireless radio channel is a difficult medium, susceptible to noise, interference, blockage, and multipath, and these channel impediments change over time because of user movement. These characteristics impose fundamental limits on the range, data rate, and reliability of communication over wireless links. These limits are determined by several factors, most significantly the propagation environment and the user mobility. For example, the radio channel for an indoor user at walking speeds typically supports higher data rates with better reliability than the channel of an outdoor user surrounded by tall buildings and moving at high speed.

Wireless systems use the atmosphere as their transmission medium. Radio signals are sent across this medium by inducing a current of sufficient amplitude in an antenna whose dimensions are approximately the same as the wavelength of the generated signal. (The signal wavelength  $\lambda = c/f$ , where  $f$  is the signal's carrier frequency and  $c$  is the speed of light.) We focus on radio waves in the UHF and SHF frequency bands, which occupy the 0.3- to 3-GHz and 3- to 30-GHz portions of the spectrum, respectively. Most terrestrial mobile systems use the UHF band, while satellite systems typically operate in the SHF band. At these frequencies the earth's curvature and the ionosphere do not affect signal propagation.

Figure 7.4 depicts a typical situation. The transmitted signal has a direct-path component between the transmitter and the receiver that is attenuated



**7.4**  
**FIGURE**

A wireless propagation scenario: the received signal has a direct-path component which may be attenuated or blocked, and other reflected, components.

or obstructed. Other components of the transmitted signal, referred to as *multipath components*, are reflected, scattered, or diffracted by surrounding objects and arrive at the receiver shifted in amplitude, phase, and time relative to the direct-path signal. The received signal may also experience interference from other users in the same frequency band. Based on this model the wireless radio channel has four main characteristics: path loss, shadowing, multipath, and interference.

Path loss determines how the average received signal power decreases with the distance between the transmitter and the receiver. Shadowing characterizes the signal attenuation due to obstructions from buildings or other objects. Multipath fading is caused by constructive and destructive combining of the multipath signal components, which causes random fluctuations in the received signal amplitude (flat-fading) as well as self-interference (intersymbol interference or frequency-selective fading). Interference characterizes the effects of other users operating in the same frequency band either in the same or another system.

## 7.2.1 Path Loss

*Path loss* is the ratio of received power to the transmitted power for a given propagation path and is a function of propagation distance. Free-space is the

simplest propagation model for path loss. In this model there is a direct-path signal component between the transmitter and the receiver, with no attenuating objects or multipath reflections. If  $P_R$  is the received signal power and  $P_T$  is the transmitted power, then in free-space propagation,

$$P_R \propto \frac{G \times P_T}{f^2 \times d^\alpha},$$

where  $f$  is the carrier frequency,  $d$  is the propagation distance,  $G$  is the power gain from the transmit and receive antennas, and  $\alpha = 2$ .

Radio waves in most wireless systems propagate through environments more complex than free space, where they are reflected, scattered, and diffracted by walls, terrain, buildings, and other objects. The full details of propagation in complex environments can be obtained by solving Maxwell's equations with boundary conditions that express the physical characteristics of the obstructing objects. Since these calculations are difficult, and often the necessary parameters are not available, approximations have been developed to characterize path loss. Frequently the simple exponential model for path loss above is used. In free-space propagation  $\alpha = 2$ , whereas in typical propagation environments  $\alpha$  ranges between 2 and 4.

For any path loss model, the received signal-to-noise ratio is  $\text{SNR} = P_R/N$  where  $N$  is the noise power. (Noise is usually modeled as Gaussian and white with constant power spectral density,  $N$ .) The BER of a wireless channel is a function of its SNR. The SNR required to meet a given BER target depends on the data rate of the channel, the communication techniques used, and the channel characteristics. Since path loss reduces SNR, it limits either the data rate or the signal range of a given communication system. Moreover, since the path loss exponent determines how quickly the signal power falls off with respect to distance, wireless channels with small path loss exponents will typically have larger coverage areas than those with large path loss exponents. Note that the path loss is inversely proportional to the square of the signal frequency so, for example, increasing the signal frequency by a factor of 10 reduces the received power and corresponding SNR by a factor of 100.

## 7.2.2

### Shadow Fading

The transmission path between a transmitter and a receiver is often blocked by hills or buildings outdoors and by furniture or walls indoors. Random signal variations due to these obstructing objects are referred to as *shadow fading*. Measurements in many environments indicate that the power, measured in decibels (dB), of a received signal subject to shadow fading follows a Gaussian

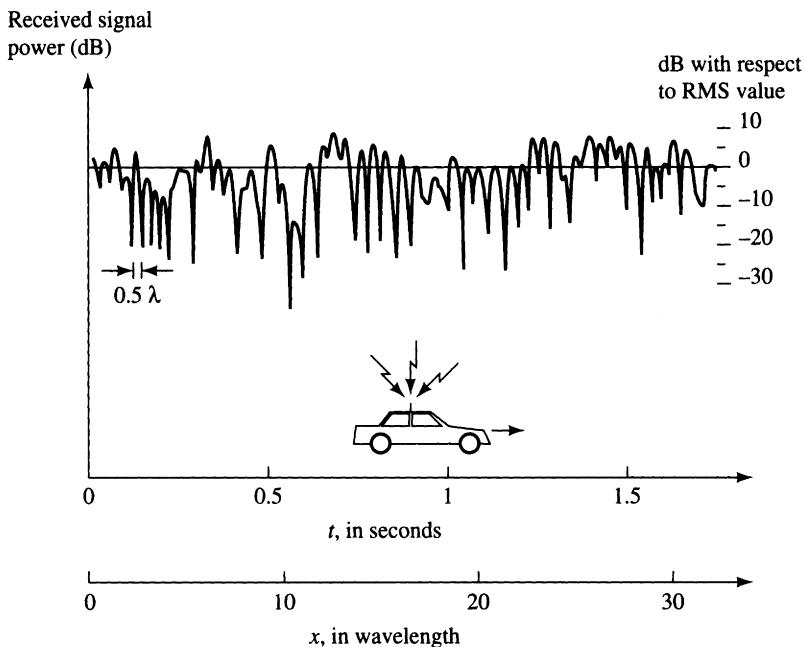
(normal) distribution, with the mean determined by path loss and the standard deviation ranging from 4 to 12 dB, depending on the environment. The random value of the shadow fading changes, or *decorrelates*, as the mobile moves past or around the obstruction.

Based on path loss alone, the received signal power at a fixed distance from the transmitter should be constant. However, shadow fading causes the received signal power at equal distances from the transmitter to be different, since some locations have more severe shadow fading than others. Thus, to ensure that the received SNR requirements are met at a given distance from the transmitter, the transmit power must be increased to compensate for severe shadow fading at some locations. This power increase imposes additional burdens on the transmitter battery and causes additional interference to other users in the same frequency band.

### 7.2.3 Multipath Flat-Fading and Intersymbol Interference

Multipath causes two significant channel impairments: *flat-fading* and *intersymbol interference*. Flat-fading describes the rapid fluctuations of the received signal power over short time periods or over short distances. Such fading is caused by the interference between different multipath signal components that arrive at the receiver at different times and hence are subject to constructive and destructive interference. This constructive and destructive interference generates a standing wave pattern of the received signal power relative to distance or, for a moving receiver, relative to time.

Figure 7.5 shows a plot of the fading exhibited by the received signal power in dB as a function of time or distance. The destructive interference can cause the received signal power to fall more than 30 dB (three orders of magnitude) below its average value. A channel is said to be in a *deep fade* whenever its received signal power falls below that required to meet the link performance specifications. Since communication links are designed with an extra power margin (link margin) of 10 to 20 dB to compensate for fading and other channel impairments, a channel is in a deep fade if its received power falls 10 to 20 dB below its average received power. Figure 7.5 shows that flat-fading channels often experience deep fades. In addition, the signal power changes drastically over distances of approximately half a signal wavelength. At a signal frequency of 900 MHz, this corresponds to every 0.3 m or every millisecond for terminals moving at 30 mph.



7.5

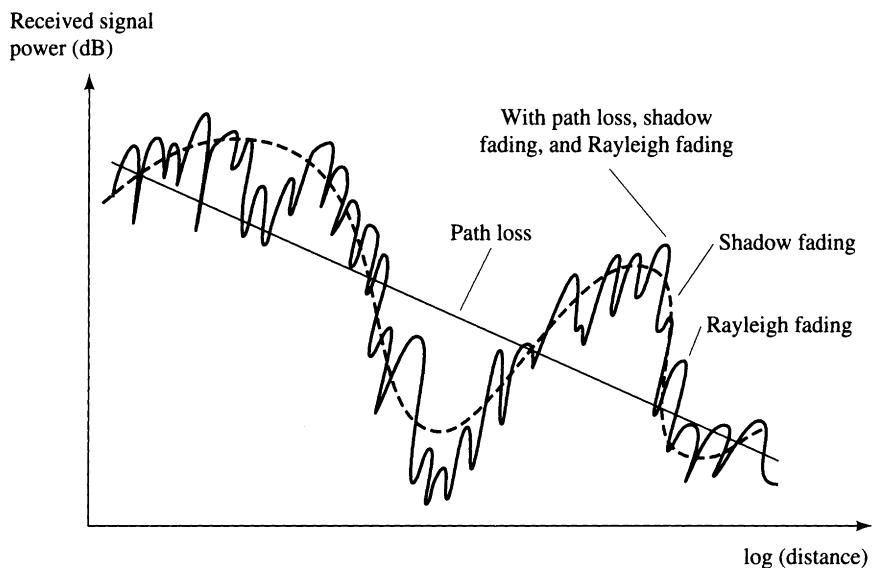
Signal fading over time and distance.

**FIGURE**

The variation in the received signal envelope of a flat-fading signal typically follows a Rayleigh distribution if the signal path between the transmitter and the receiver is obstructed and a Ricean distribution if this signal path is not obstructed. (If  $x, y$  are zero-mean, Gaussian random variables with the same variance, then  $z = x^2 + y^2)^{1/2}$  has a Rayleigh distribution. If  $m$  is a constant, then  $z = ((x + p)^2 + y^2)^{1/2}$  has a Ricean distribution.

The combination of path loss, shadowing, and flat-fading is shown in Figure 7.6. The power fall-off with distance due to path loss is fairly slow, while the signal variation due to shadowing changes more quickly, and the variation due to flat-fading is very fast. Flat-fading has two main implications for wireless link design.

First, in flat-fading the received signal power falls well below its average value. This causes a large increase in BER, which can be reduced by increasing the transmitted power of the signal. However, it is very wasteful of power to compensate for flat-fading in this manner, since deep fades occur rarely and over very short time periods. So most systems do not increase their transmit power sufficiently to remove deep signal fades. For typical user speeds and



7.6

FIGURE

Received signal power with path loss, shadow fading, and flat-fading.

data rates, these fades affect many bits, causing long strings of bit errors called *error bursts*. Error bursts are difficult to correct using error-correction codes, which typically correct only a few simultaneous bit errors. Other methods to compensate for error bursts due to flat-fading are discussed in section 7.3.3.

The other main impairment introduced by multipath is intersymbol interference (ISI). ISI becomes a significant problem when the maximum difference in the path delays of the different multipath components, called the *multipath delay spread*, exceeds a significant fraction of a bit time. The result is self-interference, since a multipath reflection carrying a given bit transmission will arrive at the receiver simultaneously with a different (delayed) multipath reflection carrying a previous bit transmission. In the frequency domain this self-interference corresponds to a nonflat frequency spectrum, so signal components at different frequencies are multiplied by different complex scale factors, thereby distorting the transmitted signal. For this reason ISI is also referred to as *frequency-selective fading*.

A channel exhibits frequency-selective fading if its *coherence bandwidth*, defined as the inverse of the channel's multipath delay spread, is less than the bandwidth of the transmitted signal. ISI causes a high BER that cannot be reduced by increasing the signal power, since that also increases the power

of the self-interference. Thus, without compensation ISI forces a reduction in data rate so that the delay spread associated with the multipath components is less than one bit time. This imposes a stringent limit on data rates, on the order of 100 Kbps for outdoor environments and 1 Mbps for indoor environments. Thus, some form of ISI compensation is needed to achieve high data rates. These compensation techniques are discussed in section 7.3.4.

#### 7.2.4 Doppler Frequency Shift

Relative motion between the transmitter and the receiver causes a shift in the frequency of the transmitted signal called the *Doppler shift*. The Doppler shift,  $f_D$ , is given by  $f_D = v/\lambda$ , where  $v$  is the relative velocity between the transmitter and the receiver and  $\lambda$  is the wavelength of the transmitted signal. Since the mobile velocity varies with time, so does the Doppler shift. This variable Doppler shift introduces an FM modulation into the signal, causing the signal bandwidth to increase by roughly  $f_D$ . For a transmit frequency of 900 MHz and a user speed of 60 mph, the Doppler shift is about 80 Hz. Since typical signal bandwidths are on the order of tens of kilohertz or more, bandwidth spreading due to Doppler is not a significant problem in most applications. However, Doppler does cause the signal to decorrelate over a time period roughly equal to  $1/f_D$ . For differential signal detection, the Doppler imposes a lower bound on the channel BER that cannot be reduced by increasing the signal power. More details on the impact of channel Doppler for differential detection are provided in section 7.3.1.

#### 7.2.5 Interference

Wireless communication channels experience interference from various sources. The main source of interference in cellular systems is frequency reuse, where frequencies are reused at spatially separated locations to increase spectral efficiency. Interference from frequency reuse can be reduced by multiuser detection, directional antennas, and dynamic channel allocation, all of which increase system complexity.

Other sources of interference in wireless systems include adjacent channel interference, caused by signals in adjacent channels with signal components outside their allocated frequency range, and narrowband interference, caused by users in other systems operating in the same frequency band. Adjacent channel interference can be mostly removed by introducing guard bands between channels, but this is wasteful of bandwidth. Narrowband interference can be removed through notch filters or spread spectrum techniques. Notch filters are

simple devices but they require knowledge of the exact location of the narrowband interference. Spread spectrum is very effective at removing narrowband interference, but it requires significant spreading of the signal bandwidth as well as an increase in system complexity. For these reasons spread spectrum is not typically used just to remove narrowband interference. Because spread spectrum allows multiple users to share the same bandwidth, it is used for multiple access.

### 7.2.6 Infrared versus Radio

In infrared communication the frequency of the transmitted signal is much higher than typical radio frequencies, around 100 GHz. Because the received signal power is inversely proportional to the square of the signal frequency, infrared transmission over large distances requires a very high transmit power or highly directional antennas. There are two main forms of infrared transmission: directive and nondirective. In directive transmission the transmit antenna is pointed directly at the receiver, whereas in nondirective transmission the signal is transmitted uniformly in all directions. Since directive transmission concentrates all its power in one direction, it achieves much higher data rates than nondirective transmission. However, these systems are severely degraded by obstructing objects and the corresponding shadow fading, which is difficult to avoid in most indoor environments with mobile users. Nondirected links have limited range due to path loss, typically in the tens of meters.

Infrared transmission enjoys a number of advantages over radio, most significantly the fact that spectrum in this frequency range is unregulated. Thus, infrared systems need not obtain an FCC license for operation. In addition, infrared systems are immune to radio interference. Infrared radiation will not penetrate walls and other opaque materials, so an infrared signal is confined to the room in which it originates. This makes infrared more secure against eavesdropping, and it allows neighboring rooms to use the same infrared links without interference. These signals are not subject to flat-fading since variations in the received signal power are integrated out by the detector.

ISI is a major problem for high-speed infrared systems, as it is for radio systems. Thus, high-speed infrared systems must use some form of ISI mitigation, typically equalization. Infrared systems are also significantly degraded by ambient light, which radiates at roughly the same frequency, causing substantial noise. In summary, infrared systems are unlikely to be used for low-cost outdoor systems because of their limited range. They have some advantages over radio systems in indoor environments but must overcome the ambient light and range limitations to be successful in these applications. Due to these

problems radio is still the dominant technology for both indoor and outdoor systems.

### 7.2.7 Capacity Limits of Wireless Channels

In 1949 Claude Shannon determined the capacity limits of communication channels with additive white Gaussian noise. For a channel without shadowing, fading, or ISI, Shannon proved that the maximum possible data rate on a given channel of bandwidth  $B$  is

$$R = B \log_2(1 + \text{SNR}) \text{ bps},$$

where SNR is the received signal-to-noise power ratio. The Shannon capacity is a theoretical limit that cannot be achieved in practice, but as link level design techniques improve, data rates for this additive white noise channel approach this theoretical bound.

The Shannon capacity for many wireless channels of interest is unknown and depends not only on the channel but also on whether the transmitter and/or the receiver can track the channel variations. For cases where the capacity is known, there is typically a large gap between actual performance and the capacity.

For example, one digital cellular standard has a 30-kHz bandwidth and a received SNR of approximately 20 dB (or more) after attenuation from shadow fading. The Shannon capacity of this channel with Rayleigh fading, assuming that the channel variation can be tracked, is on the order of 200 Kbps. However, cellular systems only achieve data rates on the order of 20 Kbps per channel. While some of this performance gap is due to channel impairments that are not incorporated into the theoretical model, most of the gap is due to the relatively inefficient signaling methods used on today's wireless channels. In wired channels, where technology is quite mature and the channel characteristics fairly benign, the data rates achieved in practice are quite close to the Shannon bound. For example, telephone loops have a capacity limit of between 30 and 60 Kbps, depending on the line quality, and modem rates today are close to this limit.

The simple formula given above for Shannon capacity is applicable to static channels with white Gaussian noise. Shannon capacity is also known for static channels with nonwhite noise and intersymbol interference. However, determining the capacity limits of time-varying wireless channels with shadowing, multipath fading, and intersymbol interference is quite challenging, and

depends on the channel characteristics, the channel rate of change, and the ability to track the channel variations.

A relatively simple lower bound for the capacity of any channel is the Shannon capacity under the worst-case propagation conditions. This can be a good bound to apply in practice, since many communication links are designed to have acceptable performance even under the worst-case conditions. But worst-case system design and capacity evaluation can be overly pessimistic. For channels with severe multipath, the channel capacity under worst-case fading conditions can be equal or close to zero. Fading compensation techniques can be used on these channels to increase both their Shannon capacity and their achievable data rates in practice. Some of these compensation techniques are discussed in the next section.

## **7.3 LINK LEVEL DESIGN**

The goal of link level design is to provide high data rates with low delays and BERs while using minimum bandwidth and transmit power. The link layer design must perform well in radio environments with fading, shadowing, multipath, and interference. Hardware constraints, such as imperfect timing and nonlinear amplifiers, must also be taken into consideration. Low-power implementations are needed, particularly for the mobile units, which have limited battery power. In addition, low-cost implementations for both transmitter and receiver are clearly desirable. Many of these properties are mutually exclusive and induce trade-offs in the choice of link level design techniques.

### **7.3.1 Modulation Techniques**

Digital modulation is the process of encoding a digital information signal into the amplitude, phase, or frequency of the transmitted signal. The encoding process affects the bandwidth of the transmitted signal and its robustness to channel impairments. In general, a modulation technique encodes several bits into one symbol, and the rate of symbol transmission determines the bandwidth of the transmitted signal. Since the signal bandwidth is determined by the symbol rate, having a large number of bits per symbol generally yields a higher data rate for a given signal bandwidth. However, the larger the number of bits per symbol, the greater the required received SNR for a given target BER.

Digital modulation techniques may be linear or nonlinear. In linear modulation the amplitude and/or phase of the transmitted signal varies linearly

with the digital modulating signal, whereas the transmitted signal amplitude is constant for nonlinear techniques.

Linear modulation techniques, including all forms of quadrature-amplitude modulation (QAM) and phase-shift-keying (PSK), use less bandwidth than nonlinear techniques, including various forms of frequency/minimum-shift-keying (FSK and MSK). Since linear techniques encode information into the amplitude and phase of linear modulation, this type of modulation is more susceptible to amplitude and phase fluctuations caused by multipath flat-fading. In addition, the amplifiers used for linear modulation must be linear, and these amplifiers are more expensive and less efficient than nonlinear amplifiers. Thus, the bandwidth efficiency of linear modulation is generally obtained at the expense of hardware cost, power, and higher BERs in fading. Linear modulation techniques are used in most wireless LAN products, whereas nonlinear techniques are used in most cellular and wide area wireless data systems.

Linear modulation techniques can be detected coherently or differentially. Coherent detection requires the receiver to obtain a coherent phase reference for the transmitted signal. This is difficult to do in a rapidly fading environment, and also increases the complexity of the receiver. Differential detection uses the previously detected symbol as a phase reference for the current symbol. Because this detected symbol is a noisy reference, differential detection requires roughly twice the power of coherent detection for the same BER. Moreover, if the channel is changing rapidly, then differential detection is not very accurate, since the channel phase may change considerably over one symbol time. As a result, rapidly changing channels with differential detection have an irreducible error floor, that is, the BER of the channel has a lower bound (error floor) that cannot be reduced by increasing the received SNR. This error floor increases as the rate of channel variation (the channel Doppler) increases and decreases as the data rate increases (since a higher data rate corresponds to a shorter bit time, so the channel phase has less time to decorrelate between bits). For high-speed wireless data (above 1 Mbps), the error floor is quite low at user speeds below 60 mph, but at lower data rates the error floor becomes significant, thereby preventing the use of differential detection.

### 7.3.2 Channel Coding and Link Layer Retransmission

Channel coding adds redundant bits to the transmitted bit stream, which are used by the receiver to correct errors introduced by the channel. This allows for a reduction in transmit power to achieve a given target BER and

also prevents packet retransmissions if all the bit errors in a packet can be corrected. Conventional forward error-correction (FEC) codes use block or convolutional code designs to produce the redundant bits for FEC; the error-correction capabilities of these code designs are obtained at the expense of increased signal bandwidth or a lower data rate. Trellis codes, invented in the early 1980s, use a joint design of the channel code and the modulation to provide FEC without bandwidth expansion or rate reduction.

The latest advance in coding technology is the family of Turbo codes, invented in 1993. Turbo codes, which achieve within a fraction of a dB of the Shannon capacity on certain channels, are complex code designs that combine an encoded version of the original bit stream with an encoded version of one or more permutations of this original bit stream. The optimal decoder for the resulting code is very complex, but Turbo codes use an iterative technique to approximate the optimal decoder with reasonable complexity. While Turbo codes exhibit dramatically improved performance over previous coding techniques and can be used with either binary or multilevel modulation, they generally have high complexity and large delays, which makes them unsuitable for many wireless applications.

Another way to compensate for the channel errors prevalent in wireless systems is to implement link layer retransmission using the Automatic Repeat Request or ARQ protocol. In ARQ data is collected into packets, and each packet is encoded with a checksum. The receiver uses the checksum to determine if one or more bits in the packet were received in error. If an error is detected, the receiver requests a retransmission of the entire packet. Link layer retransmission is wasteful, since each retransmission requires additional power and bandwidth and also interferes with other users. In addition, packet retransmissions can cause data to be delivered to the receiver out of order as well as triggering duplicate acknowledgments or end-to-end retransmissions at the transport layer, further burdening the network. While ARQ has disadvantages, in applications that require error-free packet delivery at the link layer, FEC is not sufficient and so ARQ is the only alternative.

### 7.3.3 Flat-Fading Countermeasures

The random variation in received signal power resulting from multipath flat-fading causes a very large increase in the average BER on the link. For example, in order to maintain an average BER of  $10^{-3}$  (a typical requirement for the link design of voice systems), 60 times more power is required if flat-

fading is present. (This calculation assumes binary-phase-shift-keying or BPSK modulation. Higher-level modulations require an even larger transmit power increase.) This required increase in power is even larger at the much lower BERs required for data transmission. Thus, countermeasures to combat the effects of flat-fading can significantly reduce the transmit power required on the link to achieve a target BER. The most common flat-fading countermeasures are diversity, coding and interleaving, and adaptive modulation.

In diversity, several separate, independently fading signal paths are established between the transmitter and the receiver, and the received signals obtained from each of these paths are combined. Because there is a low probability of independent fading paths experiencing deep fades simultaneously, the signal obtained by combining several such fading paths is unlikely to experience large power variations, especially with four or more diversity paths. Independent fading paths can be achieved by separating the signal in time, frequency, space, or polarization.

Time and frequency diversity are inefficient, because information is duplicated. Polarization diversity is of limited effectiveness because only two independent fading paths (corresponding to horizontal and vertical polarization) can be created, and the transmission power is usually divided between these two paths. That leaves space diversity as the most efficient diversity technique. In space diversity, independent fading paths are obtained using an antenna array, where each antenna element receives an independent fading path. In order to obtain independent fading paths, the antenna elements must be spaced at least one half signal wavelength apart, which may be difficult on small handheld devices, especially in the lower frequency bands ( $f < 1$  GHz or, equivalently,  $\lambda > .3$  m).

Another technique to combat flat-fading effects is coding and interleaving. In general, flat-fading causes bit errors to occur in bursts corresponding to the times when the channel is in a deep fade. Low-complexity channel codes can correct at most a few simultaneous bit errors, and the code performance deteriorates rapidly when errors occur in large bursts. The basic idea of coding and interleaving is to spread burst errors over many codewords. Specifically, in the interleaving process adjacent bits in a single codeword are separated by bits from other codewords and then these scrambled bits are transmitted over the channel. Since channel errors occur in bursts, the scrambling prevents adjacent bits in the same codeword from being affected by the same error burst. At the receiver the bits are deinterleaved (descrambled) back to their original order and then passed to the decoder. If the interleaving process spreads out the burst errors, and burst errors do not occur too frequently, then the codewords passed to the decoder will have at most

one bit error, and these errors are easily corrected with most FEC channel codes.

The cost of coding and interleaving (increased delay and complexity) may be large if the fading rate relative to the data rate is slow, which is typically the case for high-speed data. For example, at a channel Doppler of 10 Hz and a data rate of 10 Mbps, an error burst will last around 300,000 bits. In this case the interleaver must be large enough to handle at least that much data, and the application must be able to tolerate an interleaver delay of at least 30 ms.

When the channel can be estimated and this estimate is sent back to the transmitter, the transmission scheme can be adapted relative to the channel conditions. In particular, the data rate, power, and coding scheme can be adapted relative to the channel fading to maximize the average data rate or to minimize the average transmit power or BER. This adaptation allows the channel to be used more efficiently, since the transmission parameters are optimized to take advantage of favorable channel conditions. Several practical constraints determine when adaptive techniques should be used.

If the channel is changing so fast that it cannot be accurately estimated or the estimate cannot be fed back to the transmitter before the channel changes significantly, then adaptive techniques will perform poorly. Adaptive techniques also increase the complexity of both the transmitter and the receiver to account for the channel estimation and the adaptive transmission. Finally, a feedback path is required to relay the channel estimate back to the transmitter, which occupies a small amount of additional bandwidth on the return channel.

### 7.3.4 Intersymbol Interference Countermeasures

Techniques to combat ISI fall into two categories: signal processing and antenna solutions. Signal processing techniques, including equalization, multicarrier modulation, and spread spectrum, can either compensate for ISI at the receiver or make the transmitted signal less sensitive to ISI. Antenna solutions, including directive beams and smart antennas, change the propagation environment so that the delay between multipath components, and the corresponding ISI resulting from these delays, is reduced.

The goal of equalization is to cancel the ISI or, equivalently, to invert the effects of the channel. Channel inversion can be achieved by passing the received signal through a linear equalizing filter with a frequency response that is the inverse of the channel frequency response. This method of channel inversion is called *zero forcing*, since the ISI is forced to zero. In linear zero-

forcing equalization, the noise is also passed through the inverse channel filter, and the noise is amplified over frequencies where the channel has low gain. This noise enhancement can significantly degrade the received SNR of systems with zero-forcing equalization.

A better linear equalization technique uses an equalizing filter that minimize the average mean square error between the equalizer output and the transmitted bit stream. This type of linear equalizer is called a *minimum mean square error* equalizer. Both types of linear equalizers can be implemented in relatively simple hardware.

Although linear equalizers work well on some channels, their performance can be quite poor on channels with a long delay spread or with large variations in the channel frequency response. For these channels the nonlinear decision-feedback equalizer (DFE) tends to work much better than the linear techniques. A DFE determines the ISI from previously detected symbols and subtracts it from the incoming symbols. The DFE does not suffer from noise enhancement because it estimates the channel rather than inverting it. On most ISI channels the DFE has a much lower BER than a linear equalizer with a slightly higher complexity.

Other forms of equalization include maximum-likelihood sequence estimation and Turbo equalization, both of which usually outperform the DFE, but also have significantly higher complexity. All equalizer techniques require an accurate channel estimate, which is usually obtained by sending a training sequence over the channel. The equalizer must also track variations in the channel by periodic retraining and by adjusting the filter coefficients during data transmission based on the equalizer outputs. For this reason equalizers do not work well on channels that change so rapidly that an accurate channel estimate cannot be maintained without significant training overhead.

Multicarrier modulation is another technique to mitigate ISI. In multicarrier modulation the transmission bandwidth is divided into narrow subchannels, and the information bits are divided into an equal number of parallel streams. Each stream is used to modulate one of the subchannels, which are transmitted in parallel. Ideally the subchannel bandwidths are less than the coherence bandwidth of the channel, so that the fading on each subchannel is flat, not frequency-selective, thereby eliminating ISI. The simplest method for implementing multicarrier modulation is orthogonal (nonoverlapping) subchannels, but the spectral efficiency of multicarrier modulation can be increased by overlapping the subchannels. This is called *orthogonal frequency division multiplexing* (OFDM).

A big advantage of OFDM is that it can be efficiently implemented using the fast Fourier transform at both the transmitter and the receiver. Since

the entire bandwidth of an OFDM signal experiences frequency-selective fading, some of the OFDM subchannels will have weak SNRs. The performance over the weak subchannels can be improved by coding across subchannels, frequency equalization, or adaptive loading (transmitting at a higher data rate on the subchannels with high SNRs). The advantage of multicarrier modulation over equalization is that frequency equalization requires less training than time equalization. However, flat-fading, frequency offset, and timing mismatch impair the orthogonality of the multicarrier subchannels, resulting in self-interference that can significantly degrade performance.

Spread spectrum is a technique that spreads the transmit signal over a wide signal bandwidth in order to reduce the effects of flat-fading, ISI, and narrowband interference. In spread spectrum the information signal is modulated by a wideband pseudo-noise (PN) signal, resulting in a much larger transmit signal bandwidth than in the original signal. Spread spectrum first achieved widespread use in military applications because of its ability to hide a signal below the noise by spreading out its power over a wide bandwidth, its resistance to narrowband jamming, and its inherent ability to reduce multipath flat-fading and ISI. However, these advantages come with a significant complexity increase, especially in the receiver.

There are two common forms of spread spectrum: direct sequence, in which the data sequence is multiplied by the PN sequence, and frequency hopping, in which the narrowband signal is “hopped” over different carrier frequencies based on the PN sequence. Both techniques result in a transmit signal bandwidth that is much larger than the original signal bandwidth, hence the name *spread spectrum*.

Spread spectrum demodulation occurs in two stages: first the received signal is despread by removing the PN sequence modulation, and then the original information signal is demodulated to get the information bits. In direct sequence, despreading is accomplished by multiplying the received signal with an exact copy of the PN sequence, perfectly synchronized in time. The synchronization process entails a great deal of receiver complexity and can be degraded by interference, fading, and noise. Frequency-hopped signals are despread by synchronizing the carrier frequency at the receiver to the hopping pattern of the PN sequence.

After despreading the data signal is demodulated at baseband in the conventional way. In the despreading process, narrowband interference and delayed multipath signal components are modulated by the PN sequence, thereby spreading their signal power over the wide bandwidth of the PN sequence. The narrowband filter in the baseband demodulator then removes most of this power, which yields the narrowband interference and multipath rejection prop-

erties of spread spectrum. A RAKE receiver can also be used to coherently combine all multipath components, thereby providing improved performance through receiver diversity.

Antenna design can also reduce the effects of flat-fading and ISI. The most common wireless antenna is an omnidirectional antenna, where the power gain in all angular directions is the same. Directional antennas attenuate signals in all but a narrow angular range, and in this range signals are greatly magnified. Multipath signal components typically come from a large range of angular directions. Thus, by using directional antennas at the transmitter and/or the receiver, the power in most of the multipath components can be greatly reduced, thereby eliminating most flat-fading and ISI.

ISI reduction directly translates into increased data rates. For example, data rates exceeding 600 Mbps have been obtained experimentally in an indoor environment with directional antennas at both the transmitter and the receiver. Directional antennas can also greatly reduce interference from other users in a cellular system if these users are located outside the antenna's angular range of high power gain. However, directional antennas must be carefully positioned to point toward the user of interest, and this direction changes as users move around. This is a key motivation behind the development of steerable antennas, also called *smart* or *adaptive antennas*. These antennas change the shape and direction of their transmission beams to accommodate changes in mobile position. A steerable transmitting antenna works by controlling the phases of the signals at each of its elements, which changes the angular locations of the antenna beams (angles with large gain) and nulls (angles with small gain). Using feedback control, an antenna beam can be steered to follow the movement of a mobile, greatly reducing both flat-fading and interference from other users. Smart antennas can also provide diversity gain.

## 7.4

## CHANNEL ACCESS

Due to the scarcity of wireless spectrum, efficient techniques to share bandwidth among many heterogeneous users are needed. Applications requiring continuous transmission (e.g., voice and video) generally allocate dedicated channels for the duration of the call. Sharing bandwidth through dedicated channel allocation is called *multiple access*. In contrast to voice or video transmission, transmission of data tends to occur in bursts. Bandwidth sharing for users with bursty transmissions generally use some form of random channel

allocation that does not guarantee channel access. Bandwidth sharing using random channel allocation is called *random access*.

Whether to use multiple access or random access, and which technique to use within each access type, will depend on the traffic characteristics of the system, the state of current access technology, and compatibility with other systems. This choice is further complicated when frequencies are reused to increase spectral efficiency, as in cellular system designs. We now describe the different multiple access and random access techniques along with their corresponding trade-offs.

#### 7.4.1 Multiple Access

Multiple access techniques assign dedicated channels to multiple users through bandwidth division. Methods to divide the spectrum include frequency-division (FDMA), time-division (TDMA), code-division (CDMA), and combinations of these methods. In FDMA the total system bandwidth is divided into orthogonal channels that are nonoverlapping in frequency and are allocated to the different users. In TDMA time is divided into nonoverlapping time slots that are allocated to different users. In CDMA time and bandwidth are used simultaneously by different users, modulated by orthogonal or semi-orthogonal spreading codes. With orthogonal spreading codes the receiver can separate out the signal of interest from the other CDMA users with no residual interference between users. However, only a finite number of orthogonal spreading codes exist for any given signal bandwidth.

With semi-orthogonal spreading codes the receiver cannot completely separate out signals from different users, so that after receiver processing there is some residual interference between users. However, there is no hard limit on how many semi-orthogonal codes exist within a given signal bandwidth. This property, known as *soft capacity*, has advantages and disadvantages in the overall system design, which will be outlined in more detail below. Direct-sequence spread spectrum is often used to generate the semi-orthogonal CDMA signals. As discussed in section 7.3.4, direct-sequence spread spectrum has inherent benefits of multipath mitigation and narrowband interference rejection that are not inherent to either FDMA or TDMA, at a cost of somewhat increased complexity in the transmitter and the receiver.

FDMA is the least complex of these multiple access techniques. TDMA is somewhat more complex, since it requires timing synchronization among all users. In addition, the orthogonality of the users in TDMA is significantly degraded by ISI, since a signal transmitted in one time slot will interfere with subsequent time slots due to the multipath delay spread. Semi-orthogonal

CDMA is the most complex of the multiple access schemes due to the inherent complexity of spread spectrum in general and the additional complexity of separating out different semi-orthogonal CDMA users.

Semi-orthogonal CDMA also requires stringent power control to prevent the near-far problem. The near-far problem arises from the non-orthogonality of the spreading codes, so that every user causes interference to all other users. Due to the power falloff with distance described in section 7.2.1, the received signal power of a mobile unit located close to a receiver (or base station) is typically much larger than the received signal power of a mobile unit farther away. Thus, the interference power of this “near” mobile unit can be large in comparison with the received signal power of the “far” mobile unit. As a result the mobile units located far from the receiver typically experience high interference from other users and correspondingly poor performance. Power control mitigates the near-far problem by equalizing the received power (and the corresponding interference power) of all mobile units regardless of their distance from the receiver. However, this equalized power is difficult to maintain in a flat-fading environment and is one of the major design challenges of semi-orthogonal CDMA.

Another interesting trade-off in these multiple access methods is hard versus soft system capacity. TDMA and FDMA place a hard limit on the number of users sharing a given bandwidth, since the channel is divided into orthogonal time or frequency slots, each of which can only support one user. Orthogonal CDMA also has this hard limit. By contrast, semi-orthogonal CDMA has the advantage of soft capacity: there is no absolute limit on the number of users. However, since the semi-orthogonal codes interfere with one another, the performance of all users degrades as the number of users in the system increases; if the number of users is too large, then no user will have acceptable performance. Interference from other CDMA users can be reduced using a range of sophisticated techniques including smart antennas, interference cancellation, and multiuser detection. These techniques significantly increase the complexity of the system and can sometimes exhibit poor performance in practice.

The competing multiple access methods in the U.S. for cellular and PCS services are mixed FDMA/TDMA with three time slots per frequency channel (IS-54), semi-orthogonal CDMA (IS-95), and a combination of TDMA and slow frequency-hopping (GSM). The debate among cellular and personal communication standards committees and equipment providers over which approach to use has led to numerous analytical studies claiming superiority of one technique over the other under different channel assumptions and operating scenarios. A definitive analysis of the performance of these different multiple access techniques in all real operating environments is not possible. Thus

there is no common agreement as to which access technique is superior for any given system, especially for systems with frequency reuse or significant channel impairments.

### 7.4.2 Random Access

In most wireless data networks only a small, unpredictable, and changing subset of all the users in the network has data to send at any given time. For these systems it is inefficient to assign each user a dedicated channel. When dedicated channel access is not provided and access to the channel is not guaranteed, a random access protocol is required. Random access protocols are based on packetized data transmissions and typically fall in two categories: ALOHA techniques and reservation or demand-assignment protocols.

In pure ALOHA a transmitter will send data packets over the channel whenever data is available. This leads to a large number of data collisions at the receiver. A collision occurs when two or more packets are received simultaneously at the receiver and therefore none of the packets can be decoded correctly. Packets that collide must be resent at a later time. The throughput of an ALOHA channel, defined as the rate at which packets are correctly received, is low due to the high probability of collisions. In fact, under standard modeling assumptions, the maximum throughput in an ALOHA channel is 18% of the channel data rate (the rate that a single user could achieve were it not sharing the channel).

Fewer collisions occur if time is divided into separate slots and packet transmissions are confined to these predetermined slots, since there is no partial overlap of packets. This modification of pure ALOHA is called *slotted ALOHA*. Although slotted ALOHA roughly doubles the maximum throughput relative to pure ALOHA, this throughput is still insufficient to support bandwidth-sharing among more than a handful of high-speed users. The number of collisions in ALOHA can be reduced by the capture effect, which is similar to the near-far effect described above for spread spectrum systems. Specifically, due to the power falloff with distance of the transmitted signal, a packet transmitted from a mobile that is far from the receiver typically causes just a small amount of interference to a packet transmitted from a closer location, so that despite a collision between these packets, the latter packet can be "captured," that is, received without error. Spread spectrum can also be combined with ALOHA to reduce collisions, since when packets modulated with PN spreading sequences collide, they can be separated out by a spread spectrum receiver. As spread spectrum random access receivers must be able to demodulate the

PN spreading sequences for a large number of users, these receivers typically have very high complexity.

ALOHA with carrier sensing is often used in wired networks (e.g., the Ethernet) to avoid packet collisions. In carrier sensing a transmitter senses the channel before transmission to determine if the channel is busy. If so, then the transmission is delayed until the channel is free. Carrier sensing is often combined with collision detection, where the channel is monitored during packet transmission. If another user accesses the channel during this transmission, thereby causing a collision, then by detecting this collision the transmitter can resend the packet without waiting for a negative acknowledgment or timeout.

Carrier sensing and collision detection require a transmitting user to detect packet transmissions from other users to its intended receiver. This detection is often impossible in a wireless environment because of path loss. Suppose two transmitting users are on opposite sides of a receiver, such that the distance between each user and the receiver is half the distance between the two users. In this case, although the receiver may correctly decode a packet transmitted from either user in the absence of a collision, each user cannot detect a transmitted packet from the other user because the users are so far apart. Collision detection is also impaired by shadow fading, since the users may have an object obstructing the signal path between them. The difficulty of detecting collisions in a wireless environment is called the *hidden terminal problem* since, due to path loss and shadow fading, the signals from other users in the system may be hidden.

Since carrier sensing and collision detection are not very effective in wireless channels, the current generation of wireless LANs use *collision avoidance*. In collision avoidance the receiver notifies all nearby transmitters when it is receiving a packet by broadcasting a “busy tone.” Transmitters with packets to send wait until some random time after the busy tone terminates to begin sending their packets. The random backoff prevents all users with packets to send from simultaneously transmitting as soon as the busy tone terminates. Collision avoidance significantly increases the throughput of ALOHA and is currently part of several wireless LAN standards. However, the efficacy of collision avoidance can be degraded by the effects of path loss, shadowing, and multipath fading on the busy tone.

Reservation protocols assign channels to users on demand through a dedicated reservation channel. The channel assignment is made by a central base station or by a common algorithm running in each terminal. In such a system the total channel bandwidth is divided into data channels and reservation channels, where typically the reservation channels occupy only a small fraction of the total bandwidth. A user with data to transmit sends a short packet

containing a channel request over the reservation channel. If this packet is correctly received and a data channel is available, a data channel is reserved for the user's data transmission, and this channel assignment is conveyed back to the user. Demand-based assignment can be a very efficient means of random access since channels are only assigned when they are needed, as long as the required overhead traffic to assign channels is a small percentage of the message to be transmitted.

Several problems can arise. The setup delay and overhead associated with the channel reservation request and assignment procedure may be excessive for networks with a considerable amount of short messaging. Second, for heavily loaded systems the reservation channel may become congested with reservation requests, shifting the multiple access problem from the data channel to the reservation channel. For networks serving a wide variety of data users where a considerable amount of the network traffic consists of small messages, reservation-based random access may not be the best choice of random access protocol.

Packet-Reservation Multiple Access (PRMA) is a relatively new random access technique that combines the advantages of reservation protocols and ALOHA. In PRMA time is slotted and the time slots are organized into frames with  $N$  time slots per frame. Active terminals with packets to transmit contend for free time slots in each frame. Once a packet is successfully transmitted in a time slot, the time slot is reserved for that user in each subsequent frame, as long as the user has packets to transmit. When the user stops transmitting packets in the reserved slot the reservation is forfeited, and the user must again contend for free time slots in subsequent packet transmissions. PRMA is well suited to multimedia traffic with a mix of voice (or continuous stream) traffic and data. Once the continuous stream traffic has been successfully transmitted, it maintains a dedicated channel for the duration of its transmission, while data traffic only uses the channel as long as it is needed. PRMA requires little central control and no reservation overhead, so it is superior to reservation-based protocols when there is a mix of voice and data traffic.

All random access protocols require packet acknowledgments to guarantee successful reception of packets. If a packet is not acknowledged within some predetermined time window, then the packet is retransmitted. Since packet acknowledgments are also sent over a wireless channel, they are frequently delayed, lost, or corrupted due to channel impairments. This can result in unnecessary packet retransmission, which is inefficient, and packet duplication, which must be handled by the network layer protocol. Wireless networks can improve the likelihood of timely and uncorrupted packet acknowledgments by using more powerful link layer techniques to send these acknowledgments.

However, this requires that the link layer differentiate among different types of data transmissions, which adds to the link layer complexity.

### 7.4.3 Spectral Etiquette

Channel access allows multiple users in the same system to share a given bandwidth allocation. However, in some cases multiple systems will share the same bandwidth without any coordination, and this requires interoperability of their different access techniques and communication designs. This coexistence can be accomplished through etiquette rules, which are a minimum set of rules that allows multiple systems to share the available bandwidth fairly. These techniques offer an alternative to standardization methods that require agreement on channel access and system design before systems can be built and deployed. WINForum, an association of companies developing wireless products, has defined a set of etiquette rules for the unlicensed 2-GHz PCS bands that has been adopted by the FCC. The same set of rules is being considered for the 60-GHz spectrum allocation. The key elements of these etiquette rules are (1) listen before transmitting, to ensure that the transmitter is the only user of the spectrum while minimizing the possibility of interfering with other spectrum users; (2) limit transmission time in order to make it possible for other users to make use of the spectrum in a fair manner; and (3) limit transmitter power, so as not to interfere with users who are in nearby spectrums or reusing the same frequency spectrum some distance away.

## 7.5

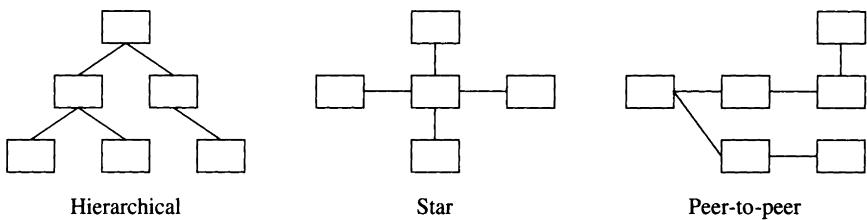
## NETWORK DESIGN

---

We first describe different network architectures and their relative trade-offs and associated distributed and centralized network control strategies. Next we consider protocols for mobility management, including the location of mobile users, user authentication, and call routing. Network reliability and QoS guarantees for wireless networks are also discussed. Some pitfalls of wired and wireless network interoperability using universal networking protocols like TCP/IP and ATM networks will be outlined, followed by a brief discussion of security.

### 7.5.1 Architecture

The three main types of network architectures are a star (central hub) topology, an ad hoc or peer-to-peer structure, and a hierarchical or tree structure. These are illustrated in Figure 7.7.



7.7  
FIGURE

Architectures for wireless networks.

Hierarchical network architectures are usually only used for wireless networks spanning a range of coverage regions, as was shown in Figure 7.1. In this figure the lowest level of the hierarchy consists of indoor systems with small coverage areas; the next level of the hierarchy consists of cellular systems covering a city, followed by systems with regional and then global coverage. Since the coverage regions define a natural hierarchy of the overall network, a hierarchical network architecture along with protocols for routing and identifying user locations are well suited to this type of system.

In a peer-to-peer architecture the nodes self-configure into an integrated network using distributed control, and the connection between any two nodes in the network consists of one or more peer-to-peer communication links. In a star architecture, communication flows from network nodes to a central hub over one set of channels, and from the hub to the nodes over a separate set of channels. The choice of a peer-to-peer or a star network architecture depends on many factors. Peer-to-peer architectures require no existing infrastructure, are easily reconfigurable, and have no single points of failure.

Peer-to-peer architectures can use multiple hops for the end-to-end connection, which has the advantage of extending the network range and the disadvantage that if one of the hops fails, the entire end-to-end connection is lost. This disadvantage is mitigated by the fact that each node may have connections to many other nodes, so there may be multiple ways to form an end-to-end connection with any other user. These advantages make peer-to-peer architectures the architecture of choice in military systems.

Since star architectures have only one hop between a network node and the central hub, they tend to be more predictable and reliable; however, if that connection is weak, then there is no alternative connection. A big advantage of star architectures is that they can use centralized control functions at the hub for channel estimation, access, routing, and resource allocation. This centralized control usually results in a more efficient and reliable network, and for this rea-

son many commercial wireless networks use the star architecture. Common examples of wireless systems with peer-to-peer architectures include packet radio networks and some wireless LANs. The star architecture is employed in cellular and paging systems.

### 7.5.2 Mobility Management

Mobility management consists of two related functions: location management and call routing. Location management is the process of identifying the physical location of the user so that calls directed to that user can be routed to that location. Location management is also responsible for verifying the authenticity of users accessing the network. Routing consists of setting up a route through the network over which data directed to a particular user is sent, and dynamically reconfiguring the route as the user location changes. In cellular systems location management and routing are coordinated by the base stations or the central mobile telephone switching office (MTSO), whereas on the Internet these functions are handled by the Mobile Internetworking Routing Protocol (Mobile IP).

The location management and routing protocols in Mobile IP and in cellular systems are somewhat different, but they both use local and remote databases for user tracking, authentication, and call routing. In cellular systems location management and call routing are handled by the MTSO in each city. An MTSO is connected to all base stations in its city via high-speed communication links. The MTSO in each city maintains a home location database for local users and a visitor location database for visiting users. Calls directed to a particular mobile unit are routed through the public-switched telephone network to the MTSO in that mobile's home city. When a mobile unit in the home city turns the handset on, that signal is relayed by the local base station to the MTSO. The MTSO authenticates the ID number of the mobile and then registers that user in its home location database. After registration, any calls addressed to that user are sent to him by the MTSO via one of its base stations.

If a mobile is roaming in a different city, turning the handset on registers the mobile with the MTSO in the visiting city. Specifically, the mobile's signal is picked up by a local base station in the visiting city, which relays the signal to the visiting city's MTSO. The visiting city's MTSO then sends a message to the MTSO in the mobile's home city requesting user authentication and call forwarding for that user. The MTSO in the mobile's home city authenticates the mobile's ID number, adds the location of the visiting city's MTSO to its home location database entry for the visiting mobile, and sends a confirmation message to the visiting city's MTSO. The visiting mobile is then registered in

the visitor location database of the visiting city's MTSO. After this process is complete, when a call for a visiting mobile arrives at that mobile's home city, the home city MTSO sets up a circuit-switched connection with the visiting city's MTSO along which the call is routed. This method of call routing is somewhat inefficient, since a call must travel from its origin to the home city's MTSO and then be rerouted to the visiting city. The MTSO also coordinates handoffs between base stations by detecting when a mobile signal is becoming weak at its current base station and finding the neighboring base station with the best connection to that mobile. This handoff process will be discussed in more detail in section 7.6.1.

Location management and routing on the Internet is handled by the Mobile IP protocol, described in section 4.3. The protocol does not support real-time handoff of a mobile between different networks: it is designed mainly for stationary users who occasionally move their computer from one network to another.

### 7.5.3 Network Reliability

An end-to-end connection in a wireless network is composed of one or more wireless and wired links, with at least one wireless link. These different links have widely varying data rates, BERs, and delays. Moreover, user mobility causes one or more of these links to change over time. These characteristics make it difficult to insure reliability of the end-to-end network connection. Protocols like TCP that are designed for wired networks do not work well in wireless networks. These protocols assume that packet losses are caused by congestion, and they react by throttling the source. On wireless networks most packet losses are due to poor link quality and intermittent connectivity. Using the congestion control mechanisms of TCP to correct for these problems can cause large and variable end-to-end delays and low network throughput. In addition, wireless channels have low data rates and high BERs, and the random characteristics of these channels make it difficult to guarantee or even predict end-to-end data rates, delay statistics, or packet loss probabilities.

Performance metrics such as data rates, end-to-end latency, and likelihood of packet loss are usually referred to as a connection's *quality of service* (QoS). The QoS requirements for a connection are based on the kind of data being transported over that connection. For example, voice has a high tolerance to packet loss but a low tolerance for delay, whereas data has the opposite requirements.

The inherent impairments and random variations of the wireless channel make it difficult to provide anything other than best-effort service in wire-

less networks. This difficulty is the main challenge in supporting high-speed real-time applications like video teleconferencing over these networks. One possible approach to compensate for the lack of QoS guarantees is to adapt at the application layer to the variable QoS offered by the network, as described in more detail in section 7.5.6.

#### 7.5.4 Internetworking

In order to connect wired and wireless networks together, they must share a common networking protocol such as TCP/IP and ATM. As we have seen, TCP has problems operating over wireless links, mainly due to its use of congestion control in response to packet delays.

However, wireless links can experience large and variable delays, sporadic error bursts, and intermittent connectivity due to handoffs. Large and variable link delays cause large oscillations in the TCP sending rate, resulting in large and variable end-to-end delays. Error bursts can result in unnecessary retransmissions by TCP, since these errors are usually corrected at the link layer, and also cause significant throughput degradation, since flow is reduced in response to every error burst. The effect of intermittent connectivity on TCP is similar to that of error bursts, resulting in unnecessary retransmissions and throughput reduction. Various modifications to TCP have been proposed to address this issue, but none has emerged as a clear solution.

ATM provides QoS guarantees, which are required for some applications. But it is not clear that the QoS guarantees of ATM can be achieved in a wireless network.

#### 7.5.5 Security

Wireless communication systems are inherently less private than wireline systems because the wireless link can be intercepted without any physical tap, and this interception cannot be detected by the transmitter or the receiver. This lack of link security also makes wireless networks more subject to usage fraud and activity monitoring than their wireline counterparts. Opportunities for fraudulent attacks will increase as services like wireless banking and commerce become available. Thus, security technology is an important challenge. Security issues can be broken down into three categories: network security, radio link security, and hardware security.

Network security includes countermeasures to fraudulent access and monitoring of network activity, and end-to-end encryption. Radio link security

entails preventing interception of the radio signal, ensuring privacy of user location information and, for military applications, anti-jam and low probability of interception and detection capabilities. Hardware security should prevent fraudulent use of the mobile terminal in the event of theft or loss, and user databases should also be secure against unauthorized access.

### 7.5.6 A New Paradigm for Wireless Network Design

Network design using the layered OSI architecture has worked well for wired networks, especially as the communication links evolved to provide gigabit-per-second data rates and BERs of  $10^{-12}$ . Wireless channels typically have much lower data rates (tens or hundreds of Kbps for typical channels with high user mobility), higher BERs ( $10^{-2}$  to  $10^{-6}$ ), and exhibit sporadic error bursts and intermittent connectivity. These performance characteristics also vary over time, as do the network topology and user traffic. Consequently, good end-to-end wireless network performance, will not be possible without a truly optimized, integrated, and adaptive network design.

In order to optimize network performance, each level in the protocol stack should adapt to wireless link variations in an appropriate manner, taking into account the adaptive strategies at the other layers. Therefore, an integrated adaptive design across all levels of the protocol stack is needed to best exploit interdependencies among protocol layers.

This integrated approach to adaptive protocol design entails two related questions: what performance metrics should be measured at each layer of the protocol stack, and what adaptive strategies should be developed for each layer of the protocol stack to best respond to variations in these local performance metrics. Network design based on the OSI model considers these two questions in isolation for each layer. But the best answer to both questions at a particular layer depends on how and to what the network adapts at other layers of the protocol stack. In other words, the best overall network design requires that these questions be addressed at all layers of the protocol stack *simultaneously*.

The integrated adaptive protocol design should still be based on a hierarchical approach, since network variations take place on different time scales. Variations in link SNR (i.e., BER and connectivity) can be very fast, on the order of microseconds for vehicle-based users. Network topology changes more slowly, on the order of seconds, while variations of user traffic may change over tens to hundreds of seconds (although this may change as networks support more applications with short messaging). The different time scales of the

network variations suggest a hierarchical approach for protocol design, since the rate at which a protocol can adapt to overall network changes is, to a large extent, determined by its location in the protocol stack.

For example, suppose the link connectivity (link SNR) in the wireless link of an end-to-end network connection is weak. By the time this connectivity information is relayed to a higher level of the protocol stack (e.g., the network layer for rerouting or the application layer for reduced-rate compression), the link SNR may change. Therefore, it makes sense for each protocol layer to adapt to variations that are local to that layer. If this local adaptation is insufficient to compensate for the local performance degradation, then the performance metrics at the next layer of the protocol stack will degrade as a result. Adaptation at this next layer may then correct or at least mitigate the problem that could not be fixed through local adaptation.

Consider the weak link situation. Link connectivity can be measured quite accurately and quickly at the link level. The link protocol can therefore respond to weak connectivity by increasing its transmit power or its error-correction coding. This will correct for variations in connectivity due, for example, to multipath flat-fading. However, if the weak link is caused by something difficult to correct at the link layer (e.g., the mobile unit is inside a tunnel), then it is better for a higher layer of the network protocol stack to respond (for instance, by delaying packet transmissions until the mobile leaves the tunnel). However, real-time applications may not be able to tolerate an increase in packet delay, in which case the application can adapt by reducing its rate of packet transmission. This may entail using a lower rate compression scheme or sending only priority data (e.g., the voice component of a video stream or the low resolution components of an image). It is this integrated approach to adaptive networking—how each layer of the protocol stack should respond to local variations given adaptation at higher layers—that should be considered as a new paradigm in wireless network design.

---

## 7.6

## WIRELESS NETWORKS TODAY

In this section we give a brief overview of wireless networks in operation today, including cellular systems, cordless phones, wireless LANs, wide area wireless data systems, paging systems, and satellite systems. These systems are mainly differentiated by their application (voice or data), support for user mobility, and coverage areas.

### 7.6.1 Cellular Telephone Systems

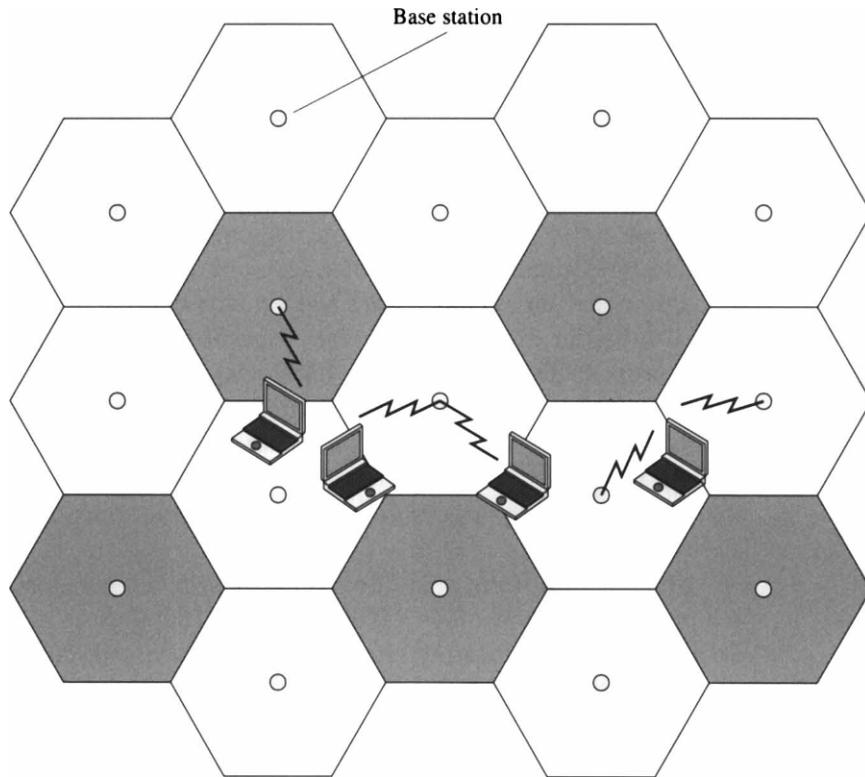
Cellular telephone systems, also referred to as Personal Communication Systems (PCS), are extremely popular and lucrative worldwide: these systems have sparked much of the optimism about the future of wireless networks. Cellular telephone systems are designed to provide two-way voice communication at vehicle speeds with regional or national coverage. Cellular systems were initially designed for mobile terminals inside vehicles with antennas mounted on the vehicle roof. Today these systems have evolved to support lightweight handheld mobile terminals operating inside and outside buildings at both pedestrian and vehicle speeds.

The basic feature of the cellular system is frequency reuse, which exploits path loss to reuse the same frequency spectrum at spatially separated locations. Specifically, the coverage area of a cellular system is divided into nonoverlapping *cells* where some set of channels is assigned to each cell. This same channel set is used in another cell some distance away, as shown in Figure 7.8, where the shaded cells use the same channel set.

Operation within a cell is controlled by a centralized base station, as described in more detail below. The interference caused by users in different cells operating on the same channel set is called *intercell interference*. The spatial separation of cells that reuse the same channel set, the *reuse distance*, should be as small as possible to maximize the spectral efficiency obtained by frequency reuse. However, as the reuse distance decreases, intercell interference increases, due to the smaller propagation distance between interfering cells. Since intercell interference must remain below a given threshold for acceptable system performance, reuse distance cannot be reduced below some minimum value. In practice it is quite difficult to determine this minimum value since both the transmitting and interfering signals experience random power variations due to path loss, shadowing, and multipath.

In order to determine the best reuse distance and base station placement, an accurate characterization of signal propagation within the cells is needed. This characterization is usually obtained using detailed analytical models, sophisticated computer-aided modeling, or empirical measurements.

Initial cellular system designs were mainly driven by the high cost of base stations, about \$1 million each. For this reason early cellular systems used a relatively small number of cells to cover an entire city or region. The cell base stations were placed on tall buildings or mountains and transmitted at very high power with cell coverage areas of several square miles. These large cells



7.8

Cellular systems with frequency reuse.

FIGURE

are called *macrocells*. Signals propagated out from base stations uniformly in all directions, so a mobile moving in a circle around the base station would have approximately constant received power. This circular contour of constant power yields a hexagonal cell shape for the system, since a hexagon is the closest shape to a circle that can cover a given area with multiple nonoverlapping cells.

Cellular telephone systems are now evolving to smaller cells with base stations close to street level or inside buildings transmitting at much lower power. These smaller cells are called *microcells* or *picocells*, depending on their size. This evolution is driven by two factors: the need for higher capacity in areas with high user density, and the reduced size and cost of a base station. A cell of any size can support roughly the same number of users if the system is scaled accordingly. Thus, for a given coverage area, a system with many

microcells has a higher number of users per unit area than a system with just a few macrocells. Small cells also have better propagation conditions since the lower base stations have reduced shadowing and multipath.

In addition, less power is required at the mobile terminals in microcellular systems, since the terminals are closer to the base stations. However, the evolution to smaller cells has complicated network design. Mobiles traverse a small cell more quickly than a large cell, and therefore handoffs must be processed more quickly. Location management also becomes more complicated, since there are more cells within a given city where a mobile may be located. It is also harder to develop general propagation models for small cells, since signal propagation in these cells is highly dependent on base station placement and the geometry of the surrounding reflectors. In particular, a hexagonal cell shape is not a good approximation to signal propagation in microcells. Microcellular systems are often designed using square or triangular cell shapes, but these shapes have a large margin of error in their approximation to microcell signal propagation.

All base stations in a city are connected via a high-speed link to a mobile telephone switching office (MTSO). The MTSO acts as a central controller for the network, allocating channels within each cell, coordinating handoffs between cells when a mobile traverses a cell boundary, and routing calls to and from mobile users in conjunction with the public-switched telephone network (PSTN). A new user located in a given cell requests a channel by sending a call request to the cell's base station over a separate control channel. The request is relayed to the MTSO, which accepts the call request if a channel is available in that cell. If no channel is available, the call request is rejected.

A call handoff is initiated when the base station or the mobile in a given cell detects that the received signal power for that call is approaching a minimum threshold. In this case the base station informs the MTSO that the mobile requires a handoff, and the MTSO then queries surrounding base stations to determine if one of these stations can detect that mobile's signal. The MTSO coordinates a handoff between the original base station and the new base station. If no channel is available in the cell with the new base station, the handoff fails and the call is terminated. False handoffs may also be initiated if a mobile is in a deep fade, causing its received signal power to drop below the minimum threshold even though it may be nowhere near a cell boundary.

Cellular telephone systems have moved from analog to digital technology. Digital technology has many advantages. The components are cheaper, faster, smaller, and require less power. Voice quality is improved due to error-correction coding. Digital systems also have higher capacity than analog systems since they are not limited to FDMA multiple access, and they can take

advantage of advanced compression techniques and voice activity factors. In addition, encryption techniques can be used to secure digital signals against eavesdropping.

All cellular systems being deployed today are digital, and these systems provide voice mail, paging, and e-mail services in addition to voice. Due to their lower cost and higher efficiency, service providers have used aggressive pricing tactics to encourage user migration from analog to digital systems. Since they are relatively new, digital systems do not always work as well as the old analog ones. Users experience poor voice quality, frequent call dropping, short battery life, and spotty coverage in certain areas. System performance will certainly improve as the technology and networks mature. However, it is unlikely that cellular phones will provide the same quality as wireline service any time soon. The great popularity of cellular systems indicates that users are willing to tolerate inferior voice communications in exchange for mobility.

Digital cellular systems can use any of the multiple access techniques described above to divide up the signal bandwidth in a given cell. In the U.S. the standards activities surrounding the current generation of digital cellular systems provoked a raging debate on multiple access for these systems, resulting in several incompatible standards. In particular, there are two standards in the 900-MHz (cellular) frequency band: IS-54, which uses a combination of TDMA and FDMA, and IS-95, which uses semi-orthogonal CDMA. The spectrum for digital cellular in the 2-GHz (PCS) frequency band was auctioned off, so service providers could use an existing standard or develop proprietary systems for their purchased spectrum.

The end result has been three different digital cellular standards for this frequency band: IS-136 (which is basically the same as IS-54 at a higher frequency), IS-95, and the European digital cellular standard GSM, which uses a combination of TDMA and slow frequency-hopping. The digital cellular standard in Japan is similar to IS-54 and IS-136 but in a different frequency band, and the GSM system in Europe is at a different frequency than the GSM systems in the U.S. This proliferation of incompatible standards makes it impossible to roam between systems nationwide or globally without using multiple phones (and phone numbers).

Efficient cellular system designs are *interference-limited*, that is, the interference dominates the noise floor since otherwise more users could be added to the system. As a result, any technique to reduce interference in cellular systems leads directly to an increase in system capacity and performance. Some methods for interference reduction in use today or proposed for future systems include cell sectorization, directional and smart antennas, multiuser detection, and dynamic channel and resource allocation.

## 7.6.2 Cordless Phones

Cordless telephones first appeared in the late 1970s and have become very popular. Almost half of the phones in U.S. homes today are cordless. Cordless phones originally provided a low-cost low-mobility wireless link replacing the cord connecting a telephone base unit and its handset. Initial cordless phones had poor voice quality and were quickly discarded by users. The first cordless systems allowed only one phone handset to connect to each base unit, and coverage was limited to a few rooms of a house or office. This is still the main purpose of cordless telephones in the U.S. today, although they now use digital technology. In Europe and the Far East, digital cordless phone systems have evolved to provide coverage over much wider areas, both in and away from home, and are similar in many ways to today's cellular telephone systems.

Digital cordless phone systems in the U.S. today consist of a wireless handset connected to a single base unit, which in turn is connected to the telephone network. These cordless phones impose no added complexity on the network, since the cordless base unit acts like a wireline telephone for networking purposes. The movement of the cordless handset is extremely limited: it must remain within range of its base unit. There is no coordination with other cordless phone systems, so a high density of these systems in a small area, such as an apartment building, can result in significant interference among systems. For this reason cordless phones today have multiple voice channels and scan among these channels to find the one with minimal interference. Spread spectrum cordless phones have also been introduced to reduce interference from other systems and to mitigate narrowband interference.

In Europe and the Far East, the second generation of digital cordless phones (CT-2, for cordless telephone, second generation) have an extended range of use beyond a single residence or office. Within a home these systems operate as conventional cordless phones. To extend the range beyond the home, base stations, also called *phone-points* or *telepoints*, are mounted in places where people congregate, like shopping malls, busy streets, train stations, and airports. Cordless phones registered with the telepoint provider can place calls whenever they are in range of a telepoint. Calls cannot be received from the telepoint since the network has no routing support for mobile users, although some newer CT-2 handsets have built-in pagers to compensate for this deficiency. These systems also do not hand off calls so a user must remain within range of the telepoint for the duration of a call.

Telepoint service was introduced twice in the United Kingdom and failed both times, but these systems grew rapidly in Hong Kong and Singapore through the mid 1990s. This rapid growth deteriorated quickly after the first

few years, as cellular phone operators cut prices to compete with telepoint service. The main complaint about telepoint service was the incomplete radio coverage and lack of handoff. Since cellular systems avoid these problems, as long as prices were competitive there was little reason for people to use telepoint services. Most of these services have now disappeared.

Another evolution of the cordless telephone designed primarily for office buildings is the European DECT system. DECT provides local mobility support for users in an in-building private branch exchange (PBX). DECT base units are mounted throughout a building, and each base station is attached through a controller to the PBX of the building. Handsets communicate to the nearest base station in the building, and calls are handed off as a user walks between base stations. DECT can also ring handsets from the closest base station. The DECT standard also supports telepoint services, although this application has not received much attention, probably due to the failure of CT-2 services. There are currently around 7 million DECT users in Europe, but the standard has not yet spread to other countries.

The most recent advance in cordless telephone system design is the Personal Handyphone System (PHS) in Japan. The PHS system is quite similar to a cellular system, with widespread base station deployment supporting hand-off and call routing between base stations. With these capabilities PHS does not suffer from the main limitations of the CT-2 system. Initially PHS systems enjoyed one of the fastest growth rates ever for a new technology. In 1997, two years after its introduction, PHS subscribers peaked at about 7 million users; this number has declined slightly since then due mainly to sharp price-cutting by cellular providers. The main difference between a PHS system and a cellular system is that PHS cannot support call handoff at vehicle speeds. This deficiency is mainly due to the dynamic channel allocation procedure used in PHS. Dynamic channel allocation greatly increases the number of handsets that can be serviced by a single base station, thereby lowering the system cost, but it also complicates the handoff procedure. It is too soon to tell if PHS systems will go the same route as CT-2. However, it is clear from the recent history of cordless phone systems that to extend the range of these systems beyond the home requires either the same functionality as cellular systems or a significantly reduced cost.

### 7.6.3 Wireless LANs

Wireless LANs provide high-speed data within a small region such as a campus or small building, as users move from place to place. Wireless devices that access these LANs are typically stationary or moving at pedestrian speeds. Nearly

all wireless LANs in the United States use one of the ISM frequency bands. The appeal of these frequency bands, located at 900 MHz, 2.4 GHz, and 5.8 GHz, is that an FCC license is not required to operate in these bands. However, this advantage is a double-edged sword, since many other systems operate in these bands for the same reason, causing a great deal of interference among systems. The FCC mitigates this interference problem by setting a stringent limit on the power per unit bandwidth for ISM-band systems. To satisfy this requirement, wireless LANs use either direct-sequence or frequency-hopping spread spectrum so that their total power is spread over a wide bandwidth. Wireless LANs can have either a star architecture, with wireless access points or hubs placed throughout the coverage region, or a peer-to-peer architecture, where the wireless terminals self-configure into a network.

Dozens of wireless LAN companies and products appeared in the early 1990s to meet the estimated demand for high-speed wireless data. These first wireless LANs were based on proprietary and incompatible protocols, although most operated in the 900-MHz ISM band using direct-sequence spread spectrum with data rates on the order of 1 to 2 Mbps. Both star and peer-to-peer architectures were used. The lack of standardization for these products led to high development costs, low-volume production, and small markets for each individual product. Of these original products, only a handful remain, including Proxim's RangeLAN, Lucent's WaveLAN, and Windata's FreePort. Only one of the first generation wireless LANs, Motorola's Altair, operated outside the 900-MHz ISM band. This system, operating in the licensed 18-GHz band, had data rates on the order of 6 Mbps. However, performance of Altair was hampered by the high cost of components and the increased path loss at 18 GHz. As a result Altair was recently discontinued.

The 900-MHz ISM band is not available in most parts of the world, so the new generation of wireless LANs operate in the 2.4-GHz ISM band, which is available worldwide. A wireless LAN standard for this frequency band, the IEEE 802.11 standard, was recently completed to avoid some of the problems with the proprietary first generation systems. The standard specifies frequency-hopped spread spectrum with data rates of 1.6 Mbps and a range of approximately 500 ft. The network architecture can be either star or peer-to-peer. Many companies have developed products based on the 802.11 standard, and these products are constantly evolving to provide higher data rates and better coverage. Cabletron's Frealink is the only existing wireless LAN in the 5.8-GHz range: this system has slightly higher data rates and slightly lower range than the 802.11 systems. Because data rates are low and coverage is limited, the market for all wireless LANs has remained relatively flat (around \$200 million, far below the billion dollar market of today's cellular systems). Optimism remains high that the wireless LAN market is poised to take off, although this prediction has been

made every year since the inception of wireless LANs yet the market has so far failed to materialize.

#### 7.6.4 Wide Area Wireless Data Services

Wide area wireless data services in the U.S. provide low rate wireless data to high-mobility users over a very large coverage area. The initial two service providers for wireless data were the ARDIS network run by Motorola and RAM Mobile Data, which uses Ericsson's Mobitex technology. ARDIS and RAM Mobile Data provide service to most metropolitan areas within the U.S. In these systems a large geographical region is serviced by a few base stations mounted on towers, rooftops, or mountains and transmitting at high power. In ARDIS the base stations are connected to network controllers attached to a backbone network, whereas in RAM Mobile Data the base stations are at the bottom of a hierarchical network architecture. Both systems use a form of the ALOHA protocol for random access with collision reduction through either a busy tone transmission or carrier sensing. Initial data rates for these systems were low, 4.8 Kbps for ARDIS and 8 Kbps for RAM, but these rates have since both increased to 19.2 Kbps.

Another provider is Metricom, with systems operating in the San Francisco Bay Area, Seattle, and Washington, D.C. The Metricom architecture is similar to that of microcell systems: a large network of small inexpensive base stations with small coverage areas mounted close to street level. The increased efficiency of microcells allows for higher data rates in Metricom, 76 Kbps, than in the other wide area wireless data systems. Metricom uses frequency-hopped spread spectrum in the 900-MHz ISM band, with power control to minimize interference and improve battery life.

The cellular digital packet data (CDPD) system is a wide area wireless data service overlaid on the analog cellular telephone network. CDPD shares the FDMA voice channels of the analog systems, since many of these channels are idle due to the growth of digital cellular. The CDPD service provides packet data transmission at rates of 19.2 Kbps and is available throughout the U.S.

These services have failed to attract as many subscribers as initially predicted, possibly because of their low data rates and high price. The services do not seem essential: voice communication on the move seems essential to many, but people apparently prefer to wait until they have access to a phone line or wired network for data exchange. This may change with the proliferation of laptop and palmtop computers and the insatiable demand for constant Internet access and e-mail exchange.

### 7.6.5 Paging Systems

Paging systems provide very low rate one-way data services to highly mobile users over a very wide coverage area. Paging systems currently serve 56 million customers in the United States. However, the popularity of paging systems is declining as cellular systems become cheaper and more ubiquitous. In order to remain competitive, paging companies have slashed prices, and few are currently profitable. To reverse their declining fortunes, a consortium of paging service providers has recently teamed up with Microsoft and Compaq to incorporate paging functionality and Internet access into palmtop computers.

Paging systems broadcast a short paging message simultaneously from many tall base stations or satellites transmitting at very high power (hundreds of watts to kilowatts). Systems with terrestrial transmitters are typically localized to a particular geographic area, such as a city or metropolitan region, while geosynchronous satellite transmitters provide national or international coverage. In both types of systems, no location management or routing functions are needed, since the paging message is broadcast over the entire coverage area. The high complexity and power of the paging transmitters allows low-complexity, low-power, pocket paging receivers with a long usage time from small and lightweight batteries. In addition, the high transmit power allows paging signals to easily penetrate building walls. Paging service also costs less than cellular service, both for the initial device and for the monthly usage charge, although this price advantage has declined considerably in recent years. The low cost, small and lightweight handsets, long battery life, and ability of paging devices to work almost anywhere indoors or outdoors are the main reasons for their appeal.

Some paging services today offer rudimentary (1 bit) answer-back capabilities from the handheld paging device. But the requirement for two-way communication destroys the asymmetrical link advantage so well exploited in paging system design. A paging handset with answer-back capability requires a modulator and transmitter with sufficient power to reach the distant base station. These requirements significantly increase the size and weight and reduce the usage time of the handheld pager. This is especially true for paging systems with satellite base stations, unless terrestrial relays are used.

### 7.6.6 Satellite Networks

Satellite systems provide voice, data, and broadcast services with widespread, often global, coverage to high-mobility users as well as to fixed sites. They have the same basic architecture as cellular systems, except that the base

stations are satellites orbiting the earth. Satellites are characterized by their orbit distance from the earth: low-earth orbit (LEO) at 500 to 2,000 km, medium-earth orbit (MEO) at 10,000 km, and geosynchronous orbit (GEO) at 35,800 km. A geosynchronous satellite has a large coverage area that is stationary over time, since the earth and satellite orbits are synchronous. Satellites with lower orbits have smaller coverage areas, and these coverage areas change over time so that satellite handoff is needed for stationary users or fixed-point service.

Since geosynchronous satellites have such large coverage areas, just a handful of satellites are needed for global coverage. However, geosynchronous systems have several disadvantages for two-way communication. It takes a great deal of power to reach these satellites, so handsets are typically large and bulky. The large round-trip propagation delay is quite noticeable in two-way voice communication. Recall that high-capacity cellular systems require small cell sizes. Since geosynchronous satellites have very large cells, these systems have small capacity, high cost, and low data rates, less than 10 Kbps. The main geosynchronous systems in operation today are the global Inmarsat system, MSAT in North America, Mobilesat in Australia, and EMS and LLM in Europe.

The trend in current satellite systems is to use the lower LEO orbits so that lightweight handheld devices can communicate with the satellites and propagation delay does not degrade voice quality. The best known of these new LEO systems are Globalstar, Iridium, and Teledesic. Both Globalstar and Iridium provide voice and data services to globally roaming mobile users at data rates under 10 Kbps. Teledesic uses 288 satellites to provide global coverage to fixed-point users at data rates up to 2 Mbps. The cell size for each satellite in a LEO system is much larger than terrestrial cells, with the corresponding decrease in capacity associated with large cells. The cost to build, launch, and maintain these satellites is much higher than that of terrestrial base stations, so these new LEO systems are unlikely to be cost-competitive with terrestrial cellular and wireless data services. LEO systems can complement terrestrial systems in low-population areas and may appeal to travelers desiring just one handset and phone number for global roaming.

### 7.6.7 Other Wireless Systems and Applications

Many other commercial systems use wireless technology. Remote sensor networks that collect data from unattended sensors and transmit the data back to a central processing location are used for indoor (equipment monitoring, climate

control) and outdoor (earthquake sensing, remote data collection) applications. Satellite systems that provide vehicle tracking and dispatching (OMNITRACs) are commercially successful. Satellite navigation systems (the Global Positioning System or GPS) are very widely used. A new wireless system for Digital Audio Broadcasting (DAB) has recently been introduced in Europe.

## **7.7 FUTURE SYSTEMS AND STANDARDS**

We describe some of the wireless systems and standards that will emerge over the next few years. The range of activity indicates that today's systems do not yet meet the expected demand for wireless and that the capabilities and technologies for future wireless systems will continue to evolve. The interest and activity in wireless networking is so intense that it is impossible to predict what systems will emerge more than a year or two into the future.

### **7.7.1 Wireless LANs**

There are two major activities in future wireless LAN development: HIPERLAN Type 1 and Wireless ATM. HIPERLAN (for high performance radio LAN), being developed in Europe, is a family of wireless LAN standards tailored to different kinds of users. The four members of the HIPERLAN family are classified as Types 1 through 4. HIPERLAN Type 1 is similar in terms of protocol support to the IEEE 802.11 wireless LAN standard, while HIPERLAN Types 2 and 3 support Wireless ATM. HIPERLAN Type 1 operates in the 5-GHz frequency band with data rates of 23 Mbps at a range of 150 feet. The network architecture is peer-to-peer, and the channel access mechanism uses a variation of ALOHA with prioritization based on the lifetime of packets. HIPERLAN Type 1 promises an order of magnitude data rate improvement over today's technology, making it competitive with 100-Mbps wired Ethernets.

Wireless ATM is a standard to extend ATM capabilities to wireless local access as well as to wireless broadband services, a significant challenge given the difficulty of supporting ATM's QoS guarantees over a wireless medium. Work on the standard is ongoing and involves, among others, the Wireless ATM working group of the ATM Forum and the Broadband Radio Access Networks group of ETSI. The Wireless ATM standard developed by this latter group will become the HIPERLAN Type 2 standard. Development of HIPERLAN standards for Types 2 and 3 has not yet started. The wireless ATM standard has two

components: the radio access technology and enhancements to the existing ATM protocol for support of mobile terminals. There are different technologies and protocols currently under consideration for both components. The success of Wireless ATM depends on the deployment of end-to-end ATM.

The FCC recently set aside 300 MHz of unlicensed spectrum around 5 GHz called the National Information Infrastructure (NII) frequency band. This allocation frees up a large unlicensed spectrum for wireless LAN applications with data rates up to several tens of megabits per second. The NII band is divided into three 100-MHz blocks with different power restrictions (and corresponding coverage areas) in each block. The middle block is designated for campus-area wireless LANs and is compatible with the 5-GHz HIPERLAN spectral allocation in Europe so that products developed in both places can be used interchangeably. The lower block is restricted to indoor use and the higher block is designated for community networks. In all three blocks the FCC has imposed minimal technical restrictions to provide maximum flexibility in system design. There is much activity to develop standards and products for these frequency bands, but no consensus has yet emerged on the best choice of system design.

### 7.7.2 Ad Hoc Wireless Networks

An ad hoc wireless network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized control. Ad hoc wireless networks were traditionally of interest to the military. Throughout the 1970s and 1980s DARPA funded much work in the design of ad hoc packet radio networks; however, the performance of these networks was somewhat disappointing.

Ad hoc wireless networking is experiencing a resurgence of interest because of new applications and improved technology. These networks are now being considered for many commercial applications, including in-home networking, wireless LANs, nomadic computing, and short-term networking for disaster relief, public events, and temporary offices. Both the IEEE 802.11 and HIPERLAN Type 1 wireless LAN standards support ad hoc wireless networking within a small area, and wider area networks are currently under development.

Ad hoc networks require a peer-to-peer architecture, and the topology of the network depends on the location of the different users, which changes over time. In addition, since the propagation range of a given mobile is limited, the mobile may need to enlist the aid of other mobiles in forwarding a packet to its final destination. Thus the end-to-end connection between any two mobile hosts may consist of multiple wireless hops. It is a significant technical

challenge to provide reliable high-speed end-to-end communications in ad hoc wireless networks given their dynamic network topology, decentralized control, and multihop connections.

Current research in ad hoc wireless network design is focused on distributed routing. Every mobile host in a wireless ad hoc network must operate as a router in order to maintain connectivity information and forward packets from other mobiles. Routing protocols designed for wired networks are not appropriate for this task, since they either lack the ability to quickly reflect the changing topology, or may require excessive overhead. Proposed approaches to distributed routing that quickly adapt to changing network topology without excessive overhead include dynamic source and associativity-based routing. Other protocols that address some of the difficulties in supporting multimedia applications over ad hoc wireless networks include rate-adaptive compression, power control, and resource allocation through radio clustering.

### 7.7.3 IMT-2000

International Mobile Telecommunications 2000 (IMT-2000) is a worldwide standard sponsored by the ITU. In 1992 the ITU set aside 230 MHz of global spectrum in the 2-GHz frequency band to provide wireless access to the global telecommunications infrastructure through both satellite and land-based systems. The initial goal of this spectral allocation was to facilitate a move from the worldwide collection of different second-generation wireless systems, each with its own set of standards, services, coverage areas, and spectral allocations, to a worldwide standard serving fixed and mobile users in both public and private networks.

IMT-2000 is designed to support a wide range of services including voice, high-rate and variable-rate data, and multimedia in both indoor and outdoor environments. A family of radio protocols suitable for a range of environments and applications is being developed to support these requirements. The goal for the protocol family is to maximize commonality within the family while maintaining flexibility to adapt to the different environments and applications.

The IMT-2000 standard specifies data rates of 2 Mbps for local coverage and 384 Kbps for wide area coverage, and these rates can be variable depending on link and network conditions. It also supports both continuous stream and packet data. The requirement for packet data support is the main differentiator between IMT-2000 and digital cellular systems. IMT-2000 does not require backward compatibility with current wireless systems, so the design of the radio access and networking protocols can incorporate new technologies and innovations.

A majority of the proposals for the IMT-2000 radio transmission technology submitted from Asia, Europe, and North America are based on wideband CDMA technology; and it appears that some form of this technology will be adopted as the radio access standard. Advantages of wideband CDMA include its capacity and coverage gain from frequency diversity, its ease of use in packet data transfer, its flexibility for different services, its asynchronous operation, and its built-in support for adaptive antenna arrays, multiuser detection, hierarchical cell structures, and transmitter diversity. Standardization of the IMT-2000 networking protocol is still several years away.

#### 7.7.4 High-Speed Digital Cellular

All digital cellular standards are undergoing enhancements to support high rate packet data transmission. The goal is to support the IMT-2000 data rate specification of 384 Kbps over wide areas. In the near term, GSM systems will provide data rates of up to 100 Kbps by aggregating all time slots together for a single user. The enhancement to support 384 Kbps, called Enhanced Data Services for GSM Evolution (EDGE), increases the data rate further by using a high-level modulation format combined with FEC coding. This modulation is more sensitive to fading effects, and EDGE uses adaptive modulation and coding to mitigate this problem. Specifically, EDGE defines six different modulation and coding combinations, each optimized to a different value of received SNR. The received SNR is measured at the receiver and fed back to the transmitter, and the best modulation and coding combination for this SNR value is used.

The IS-54 and IS-136 systems currently provide data rates of 40 to 60 Kbps by aggregating time slots and using high-level modulation. These TDMA standards will support 384 Kbps by migrating to the GSM EDGE standard. This new TDMA standard is referred to as IS-136HS (high-speed). The IS-95 systems will support higher data rates by evolving to the wideband CDMA standard in IMT-2000. However, it is not yet clear if that standard will be backward-compatible with the IS-95 systems.

#### 7.7.5 Fixed Wireless Access

Fixed wireless access provides wireless communications between a fixed access point and multiple terminals. These systems were initially proposed to support interactive video service to the home, but the application emphasis has now shifted to providing high-speed data access (tens of Mbps) for homes and

businesses. In the U.S. two frequency bands have been set aside for these systems: part of the 28-GHz spectrum is allocated for local distribution systems (local multipoint distribution systems or LMDS), and a band in the 2-GHz spectrum is allocated for metropolitan distribution systems (multichannel multipoint distribution services or MMDS). LMDS represents a quick means for new service providers to enter the already stiff competition among wireless and wireline broadband service providers. MMDS is a television and telecommunication delivery system with transmission ranges of 30 to 50 km. MMDS has the capability to deliver more than 100 digital video TV channels along with telephony and access to emerging interactive services such as the Internet. MMDS will mainly compete with existing cable and satellite systems.

### 7.7.6 HomeRF and Bluetooth

HomeRF is an RF standard in the 2-GHz frequency band for wireless home networking. The standard was initiated by Intel, HP, Microsoft, Compaq, and IBM to enable communications and Internet connectivity among different electronic devices in and around the home, including PCs, laptops, smart pads, and intelligent home appliances. The data rate for HomeRF is specified as 2 Mbps, with simultaneous support for voice and data, at a range of 50 meters. The HomeRF standard is expected to be finalized sometime in 1999, with products incorporating the standard introduced sometime in the year 2000.

Bluetooth is a cable-replacement RF technology for short-range connections (less than 10 meters) between wireless devices. Its main application is to connect digital cellular phones, laptop and palmtop computers, portable printers and projectors, network access points, and other portable devices without the need to carry or connect cables. The Bluetooth standard was initiated by Ericsson, IBM, Intel, Nokia, and Toshiba, and has since been adopted by over 200 telecommunications and computer companies. Products compatible with Bluetooth should appear in late 1999. The system operates in the 2.4-GHz frequency band with data rates of 700 Kbps for data and up to three voice connections at 64 Kbps.

## 7.8

## SUMMARY

A desire for mobility coupled with the demand for voice, Internet, and multimedia services indicates a bright future for wireless networks. Digital cellular and paging systems have enjoyed enormous growth, but current products and services for wireless data have not lived up to expectations. This is due mainly to their high cost and poor performance. New standards and systems are emerging

worldwide to address these performance and cost issues. These systems support a wide range of voice, data, and multimedia services for fixed and mobile users both indoors and out, in cities, rural areas, and remote regions.

There are many technical challenges to overcome in building high-performance wireless networks. The wireless channel is a difficult communications medium. Sophisticated techniques exist to compensate for many of the channel impairments, but these can entail significant cost and complexity. The spectrum must also be used extremely efficiently through advanced link layer, access, and cellular system design. Networking protocols to support roaming users and end-to-end QoS guarantees also pose a significant technical challenge. The limited size and battery life of mobile terminals impose significant complexity constraints, so complexity must be distributed throughout the network to compensate for this limitation. Finally, the unpredictable nature of the wireless channel requires adaptation across all levels of the wireless network design: the link layer, network layer, transport layer, and application layer. This requires interaction among these layers, which violates the traditional network design paradigm of designing each layer in the OSI model independently from the others. While this paradigm has worked well on wired networks, especially as wired technology has evolved to the high performance of today's networks, high-performance wireless networks will not be possible without significant technical breakthroughs at all levels of the system design as well as an integrated and adaptable design for the overall network.

---

## 7.9

## NOTES

The wireless channel has been characterized in many books and articles over the last 30 years: [R96, P92] and the references therein describe the basic models for both indoor and outdoor systems. A tutorial on wireless infrared communication systems can be found in [KB97].

Techniques for reducing interference from frequency reuse are presented in [V98, Wi98, KN96]. For discussions on Trellis and Turbo codes see [U82, BGT93, HW99, RW98]. Equalization techniques are presented in [BP79, F72, DJB95]. Orthogonal frequency-division multiplexing (OFDM) is discussed in [C85, B90].

Recent work on the capacity of wireless channels is summarized in [GC97, BPS98]. Link level design for wireless channels is treated in many textbooks [Pr95, R96, St96]. Spread spectrum for both ISI mitigation and multiple access is described in [D94, V95].

The ALOHA protocol is reviewed in [Ab96, BG92]. An informative survey of routing is [RS96]. Performance of TCP over wireless links is discussed in [BPSK97]. Early work in packet radio is reviewed in [LNT87]. For routing in ad hoc networks see [ABB96, JM96].

A special issue of the *IEEE Personal Communications Magazine* is devoted to smart antennas and their use in wireless systems [PC98]. Several textbooks provide additional details on multiple and random access for wireless networks [R96, St96, PL95, BG92]. Wireless network design is still an active area of research, and there is no definitive reference for this field.

The cellular concept is explained in [M79]. More details on the cellular phone systems and wide area wireless data services can be found in [PL95]. Emerging satellite systems are described in [AS96]. Future systems and standards are continually evolving: the best source of information in this area are the standards bodies and companies building the systems [ALM98].

## 7.10

## PROBLEMS

1. Consider a path loss model

$$\frac{P_R}{P_T} = \frac{d_0}{d},$$

where  $P_R$  is the received signal power,  $P_T$  is the transmitted signal power,  $d_0 = 100$  meters is a propagation constant,  $d$  is the propagation distance, and  $\alpha$  is the path loss exponent. If  $P_T = 100$  milliwatts (mW) and  $P_R = 1$  mW is required for acceptable performance, what is the maximum transmission range of our system for a path loss exponent  $\alpha = 2$ ? By how much would the transmission range decrease if the path loss exponent was  $\alpha = 4$ ?

2. Shadow fading often follows a log-normal distribution, which means that the dB value of the received power  $10 \log_{10}(P_R)$  follows a Gaussian distribution. Suppose the received signal power follows this log-normal distribution. Assume the average value of  $10 \log_{10}(P_R/N)$  equals 10 dB and its variance equals 4 dB, where  $N$  is assumed to be constant. What is the probability that the received SNR is less than 5 dB?
3. Consider a channel with a multipath delay spread of  $10 \mu\text{s}$ . Suppose a voice signal with a signal bandwidth of 30 kHz is transmitted over this channel. Will the channel exhibit flat or frequency-selective fading? How about for a data signal with a 1-MHz signal bandwidth?

4. The BER of binary phase-shift-keying with differential detection in white Gaussian noise (i.e., no fading, shadowing, or ISI) is  $.5e^{-\gamma}$ , where  $\gamma$  is the average received SNR. If there is also Rayleigh fading then the BER becomes  $[2(1 + \gamma)]^{-1}$ . Consider a data system with a required BER of  $10^{-6}$ . What average SNR is required to achieve this target BER both with and without Rayleigh fading?
5. The error floor due to channel Doppler for binary phase-shift-keying with differential detection is  $.5(\pi f_D T_b)^2$ , where  $f_D$  is the channel Doppler and  $T_b$  is the bit time (the inverse of the data rate). For  $f_D = 80$  Hz and a data rate of 20 Kbps, what is the error floor? How about for  $f_D = 80$  Hz and a data rate of 1 Mbps? Why does the error floor decrease as the data rate increases?
6. How many independent diversity paths can be obtained using an antenna array mounted on a laptop of length .1 meter, assuming a carrier frequency of 5 GHz?
7. Consider a wireless system with total bandwidth of 3 MHz. How many users can share this channel using FDMA, where each user is assigned a 30-kHz channel? Suppose instead we use semi-orthogonal CDMA for multiple access. If each user has a received signal power of  $P$  and the interference power caused by that user to other users is  $.01P$ , for a required signal-to-interference power ratio of 10, how many users can be accommodated in this system?
8. Radio signals travel at the speed of light, equal to  $3 \times 10^8$  meters per second. What is the maximum orbit distance of a satellite such that the round-trip propagation delay of a signal does not exceed the 100-ms delay constraint of voice systems. Based on this calculation, determine which of the three satellite orbit distances, GEO, LEO, and MEO, can support voice services.
9. The IEEE 802.11 wireless LAN standard supports both star and peer-to-peer architectures. Describe a wireless LAN application that is well suited to each type of architecture.
10. Discuss the differences between ad hoc wireless networks and cellular networks in terms of network architecture and mobility management. What are the advantages and disadvantages of each network design? What applications are best suited to each type of network?
11. Spectrum for new wireless systems is being allocated by the FCC at higher frequencies (e.g., 5 GHz and 28 GHz) than in existing systems. What are the advantages and disadvantages of building systems at these higher frequencies?

# Control of Networks

In the preceding chapters we described the trends in packet- and circuit-switched networks that, together, culminate in the design of ATM networks and in improvements of the Internet protocols. We saw in Chapter 6 that a network that combines an ATM transport layer, or an IP layer with some form of quality of service or class of service, with a high-speed physical layer such as SONET can potentially provide the large range of quality of service (QoS) necessary to support most applications. However, in order to provide this range of QoS, the network's resources (bandwidth and buffers) must be properly managed or controlled.

In this chapter you will learn the concepts and fundamental techniques used to control circuit-switched, datagram, and ATM networks in order to achieve efficient use of network resources. You will understand how different control techniques affect different network performance measures, and you will acquire the skills to evaluate those performance measures, including blocking probability, delay, and loss. You will learn the deterministic proposals of the ATM Forum for admitting new connections and for allocating resources to those connections. You will see that those proposals are based on worst-case scenarios and that more sophisticated proposals based on effective bandwidth can realize the gains from statistical multiplexing. From Chapter 2 you already know the requirements imposed by various applications; now you will be able to calculate how well a specific application can be supported by the bandwidth and buffers allocated by the network for that application.

In section 8.1 we describe the objectives and the methods of control. We explain that with good control strategies the network can carry more traffic with the same quality of service. We discuss the meaning of quality of service, and we

compare deterministic and statistical guarantees. We then classify the methods available to control the operations of circuit-switched, packet-switched, and virtual circuit-switched networks. There are four principal methods: admission control, routing, flow and congestion control, and resource allocation. A fifth method, control via pricing, is discussed in Chapter 10.

In section 8.2 we explain admission control and routing mechanisms for circuit-switched networks such as the telephone network. We point out the trade-off between accepting all calls that can be carried and the negative impact on future calls. A compromise is achieved by trunk reservations.

We discuss flow- and congestion-control procedures for datagram networks in section 8.3. Flow-control procedures attempt to prevent a source from overwhelming a destination. Typically, the destination informs the source when it is becoming congested, and the source stops transmitting. The congestion-control mechanism is used to prevent some internal nodes of the network from becoming congested. Datagram networks, such as the Internet, do not guarantee delay or throughput. Their congestion control attempts to reduce the average delay per packet for any given throughput.

Section 8.4 introduces the formulations and some of the results on the control of ATM networks. Since ATM networks use virtual circuits and guarantee delays, throughput, and loss rates of individual connections, these networks must exercise admission control, and they must reserve bandwidth and buffer capacity for connections. One key question therefore is to determine how much of these resources the network must reserve. Resource-allocation procedures determine which cells the network nodes should buffer or transmit. For instance, a node may transmit cells of a video connection before e-mail cells. The node may also discard audio cells to make room for data cells.

Chapter 9 carries out the mathematical analysis that justifies the results we describe here.

Some of the material in this chapter relies on concepts of probability theory. Although we have attempted to provide intuitive discussion, readers with no probability background may not fully appreciate the argument in some places. Those readers can skim that material.

## 8.1 OBJECTIVES AND METHODS OF CONTROL

We provide an overview of the objectives of control and the principal means available or likely to be available to exercise control. These means involve tak-

ing decisions at very different time scales and based on different information. Some examples are given to illustrate the ideas.

### 8.1.1 Overview

Let us consider an ATM network that transfers information between users in cell streams over virtual circuits. A virtual circuit specifies a route of links and switches that connects the users. The cells in different virtual circuits share the transmission bandwidth and buffers that their routes have in common. The way in which those resources are shared is determined by the network's control strategy.

Because of fluctuations in the cell streams, there are periods of time when the cells arrive in a buffer faster than the transmitter empties that buffer. When this situation occurs, the buffer stores the excess cells. This temporary storage results in delays and, in case the buffer is full, in cell loss. Delays and losses affect the quality of service provided to the user.

With better control strategies, the network can carry more virtual circuits while maintaining the quality of service promised to the users. This possibility has long been known in the case of the telephone network, where improved routing algorithms enable the network to carry more phone calls with the same blocking probability. Thus, by improving the control algorithms, network managers can provide better service without additional hardware. In a public network that sells services to users, better control leads to greater revenue. In a private network, better control results in lower cost for service.

There are four basic means of control: admission control, routing, flow and congestion control, and allocation control. As indicated in Table 8.1, the means available depend on whether the transport method used by the network is circuit, datagram, or virtual circuit switching.

### 8.1.2 Control Methods

*Admission control* determines which circuit or virtual circuit connection requests are accepted by the network. This is similar to the admission of telephone calls by the telephone network. When a subscriber places a request for a new connection or call, the network can notify the user that it is too busy to accept that request. The subscriber can then decide to place the request at some later time or to use a competitor's network, if one is available. Normally, a datagram network always accepts the packets (datagrams) submitted by a user and does not exercise admission control. A number of issues are related to the admission of calls. For instance, a user may ask the network to schedule a connection at some later time. A user may ask the network to call back when

Network type	Admission	Routing	Flow	Allocation
Circuit-switched	Free paths and costs	<ul style="list-style-type: none"> <li>• Static (list)</li> <li>• Dynamic (least full)</li> </ul>	None	None
Datagram	None	<ul style="list-style-type: none"> <li>• Static (random)</li> <li>• Dynamic (shortest)</li> </ul>	<ul style="list-style-type: none"> <li>• Link window</li> <li>• End-to-end window</li> </ul>	
Virtual circuit	Current VCs	Dynamic (largest spare capacity)	<ul style="list-style-type: none"> <li>• Window</li> <li>• Rate</li> </ul>	Priority, fairness, and QoS based

8.1

Means of control of different types of network.

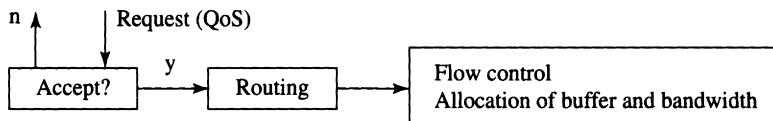
TABLE

it can set up a less expensive connection. The problems of multiparty connections, such as conference calls, are also related to admission control. Can new parties join in? What happens if one party drops out?

When the network accepts a circuit or virtual circuit call request, it must decide the path or route to be followed by the bit stream of the call or the packets. This decision is called *routing* and remains fixed for the duration of the connection. In multiparty connections, the network must decide where to copy a stream it delivers to multiple destinations. What if different destinations require different versions of the packet stream? What if the characteristics of the path from the source to the destination change over time, as in mobile wireless networks?

For datagram and virtual circuit switching, the network can also decide whether bit streams should be forwarded along their routes quickly in order to reduce delay or whether they should be slowed down (throttled) in order to prevent congestion downstream. This decision is called *congestion control*. A related method, called *traffic shaping*, is used by a traffic source to regulate its stream of packets before sending them to the network. Because a circuit-switched network provides a constant bandwidth (and no buffers) to each connection, it does not exercise congestion control.

Lastly, in virtual circuit switching, the network can control the bandwidth and buffers allocated to each virtual circuit. This is called (resource) *allocation control*. The allocation can be static, that is, fixed at the beginning of the call request, or it can be dynamic and changed over the duration of the call. It is this flexibility that permits the network to provide connections with different qualities of service. Circuit-switched networks do not exercise allocation control.



Notes: No admission control in datagram networks.

No flow control or dynamic allocation in circuit-switched networks.

### 8.1

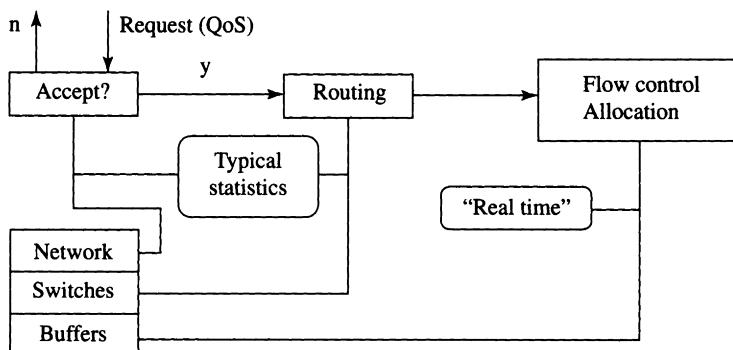
Sequence of control actions for a given call.

**FIGURE**

We will return to discuss the remaining entries of Table 8.1. Figure 8.1 illustrates the order in which the control decisions are taken. The figure is almost self-explanatory. In the case of a virtual circuit network, for example, when a request for a connection with a particular QoS is received, the network must first decide to accept or reject it. If the answer is yes, a route must then be assigned. Moreover, during the lifetime of the call, the network will exercise flow and congestion control and decide what resources to allocate to the virtual circuit.

### 8.1.3 Time Scales

The four types of control decisions are taken at different time scales, and they are based on different types of information. We illustrate this for virtual circuit networks using Figure 8.2. The decision to accept a new call is based on the



### 8.2

The different control actions are taken at different levels and based on statistics that are relevant on different time scales.

**FIGURE**

number of virtual circuits currently active in the network and on the typical statistics of the bit streams carried by these virtual circuits. The point is that the admission-control decision should not be based on the instantaneous traffic conditions; rather, it must be based on the typical conditions that are likely to prevail for the duration of the calls. Indeed, the decision to accept a call is irreversible and cannot be revoked if currently active calls suddenly become more busy than they were when the call was accepted.

The same considerations apply to routing decisions, and so these must also be based on typical traffic statistics. Thus, call admission and routing are long-term decisions that have to be maintained for the duration of the calls. By contrast, flow- and allocation-control decisions can be based on instantaneous conditions and can be modified as these conditions change; hence the label “real time” in Figure 8.2. Voice calls, for example, typically last several minutes. If voice is transmitted over a high-speed ATM network, the voice cells must be switched within a few microseconds. Thus the time scales involved in the different control actions range over several orders of magnitude.

### 8.1.4

### Examples

We now return to Table 8.1, discussing each row in turn. For a circuit-switched network, the decision to admit a new call is based on the paths that are then free or, equivalently, on the circuits that are busy (not free) when the call request is made. Thus, the network keeps track of which circuits are busy and uses that information to decide whether to accept a new call. The routing is also based on the circuits that are busy when the new call is placed. There are two types of routing algorithms: static and dynamic. A static algorithm uses precomputed decisions whereas a dynamic algorithm bases its decision on the current state of the network. There is no congestion and flow control and no allocation control in these networks, as we noted before.

A datagram network normally accepts all packets, so there is no admission control. The routing algorithm can be static or dynamic. The flow and congestion control in a datagram network typically use windows, as explained in Chapter 2. The windows may correspond to individual links or to end-to-end paths through the network. For instance, the HDLC (High-Level Data Link Control) used by X.25 networks limits the number of unacknowledged packets between two successive nodes. This is a form of link-level window congestion control. Specifically, HDLC uses the Go Back N protocol. The Selective Repeat (SRP) and Automatic Request (ARQ) protocols are other window congestion control protocols. End-to-end window retransmission protocols provide a simple way to implement flow control. By limiting the window size to a value that

the destination advertises, the source stops transmitting when the destination tells it to do so.

The TCP (Transmission Control Protocol) of the Internet uses an end-to-end window flow and congestion control (called SRP). Some datagram networks use a crude form of allocation control. For instance, TCP has a provision for expedited data transfer. A packet sent as expedited data is made to jump to the head of the queues that it goes through in the sending and receiving computers (not in the routers, since they are unaware of TCP and know only IP). More sophisticated resource allocation mechanisms have been designed and are implemented in some IP routers. We discuss these mechanisms in section 8.3.4.

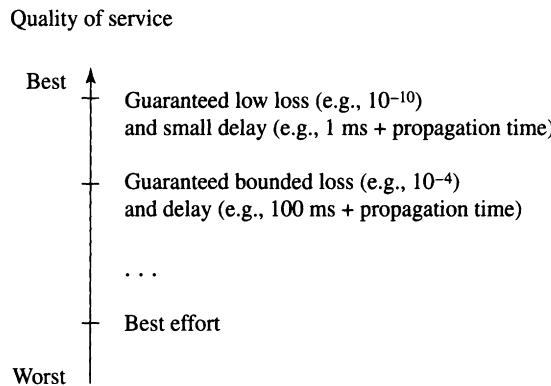
A store-and-forward network can also be used as a virtual circuit network (not separately indicated in Table 8.1). IBM's SNA (System Network Architecture) is an example of a virtual circuit store-and-forward network. Such a network accepts all requests for virtual circuits, so there is no admission control. The routing decisions are made for individual virtual circuits rather than for each packet, as in datagram networks. The networks use window flow-control methods, as in datagram networks. These networks also use a priority bandwidth allocation for expedited data packets.

ATM is the most important example of a virtual circuit-switched network. The QoS requirements in an ATM network can be more stringent than in datagram networks, and so an ATM network does not accept all virtual circuit requests. Were it to accept all the virtual circuit requests, it would be unable to deliver the cells with small loss and delay. Thus, ATM networks must control the admission of new virtual circuits. Methods are designed for such admission control and also for routing, congestion and flow control, and allocation. We will discuss these methods in section 8.4.

Another set of questions arises when different types of networks are interconnected. For instance, consider a datagram network attached to a virtual circuit network. If individual packets are sent from the datagram network to users through the virtual circuit network, then a virtual circuit needs to be set up for transporting the packets in the second network, as we explained in our discussion of IP over ATM in Chapter 6. If some form of quality of service is provided by the different networks, then they should collaborate to provide an end-to-end quality of service.

### 8.1.5 Quality of Service

We said earlier that with better control strategies the network can provide more connections or calls with the same quality of service. We will make the concept of QoS more precise.



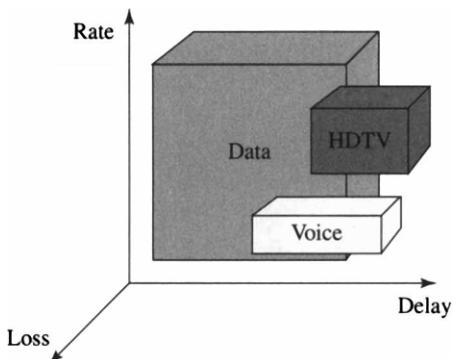
**8.3**  
**FIGURE**

Future networks will be able to offer a wide range of qualities of service from low loss and low delay to best effort.

In the best case, the QoS guarantees very small cell (or packet) loss rate and delay. By very small cell loss rate, we mean a loss rate comparable to the loss rate due to unavoidable transmission errors. For example, suppose the bit error rate along the fibers is about  $10^{-12}$  and that a cell has 424 bits (53 bytes). Then the fraction of cells lost because of transmission errors is on the order of  $424 \times 10^{-12} \approx 10^{-10}$ . This is the least cell loss rate that could be promised to users. The smallest delay that could be promised would be comparable to the propagation delay. For example, the propagation time of a cell from San Francisco to Boston is on the order of 10 ms. Thus, in the best case, the network could promise a cell loss rate of about  $10^{-10}$  and a delay on the order of 10 ms for cells going from San Francisco to Boston.

The network could propose inferior QoS. The worst quality is the so-called best-effort service, where the network promises to deliver the cells only if it finds the resources to do so. The range of QoS can be arranged from best to worst as in Figure 8.3. These different levels of QoS are similar to the different grades of service offered by the postal system. The users can choose overnight delivery, express mail, first-class mail, and so on, down to the lowest grade of bulk mail. The user selects the QoS based on the application. For instance, a user mailing a large number of advertisements might be satisfied if 95% of the customers receive them within a few weeks.

Just as in the postal system, users will select from the menu of QoS offered by the network that service which best meets the needs of their application. In Figure 8.4, the needs of several applications are displayed in a QoS parameter space of three dimensions: rate (bandwidth), loss, and delay. There are two



8.4

FIGURE

The quality of service specifies a number of parameters for the service. The figure illustrates three of the parameters and sketches their acceptable ranges of values for different types of services.

points to remember. First, quality of service can have many aspects: in addition to the three listed in the figure, one may include security, reliability, and availability of the connection. However, rate, loss, and delay are the critical aspects for most applications. Second, the network must be controlled so that it will provide different QoS to different virtual circuits (users or applications) at the same time. Providing the best QoS for all the connections is wasteful, like sending all junk mail by express delivery, and it is therefore important that the network be able to handle different connections differently. The benefits of supporting many services on one common network (due to economies of scale and scope and network externality) appear to outweigh the costs of the increased complexity.

The guarantee of quality of service across a set of interconnected networks is an important but complicated issue. In a first approximation, the rate through a series of networks is the minimum of the rates, the delay is the sum of the delays through the individual networks, and the loss rate is also approximately the sum of the individual loss rates. The actual situation is more complicated because of the needed interfaces between networks and because the delay, loss, and rate are not measured by scalar quantities.

It is useful to think of the relation between the network and a user as being ideally defined by a (service) contract. The contract obligates the network to transfer the user's information with a defined QoS (delay, loss, rate, etc.) provided that the user's traffic conforms to specified limits (bit rate, burstiness, etc.). Thus the contract says that the network will provide a particular QoS

provided the user's traffic conforms to specified conditions. With this view, the objective of network control is to fulfill the largest set of contracts.

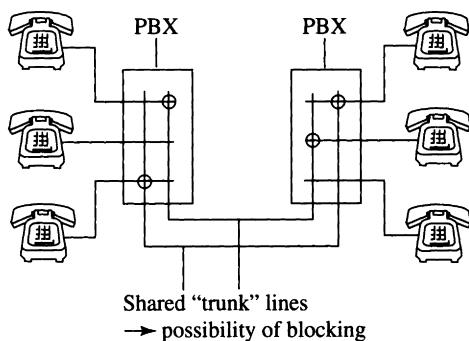
## 8.2 CIRCUIT-SWITCHED NETWORKS

We first explain, using a very simple network, that the most important QoS for circuit-switched networks is the blocking probability. We then show how the blocking probability for circuit-switched networks depends on routing and admission control.

### 8.2.1 Blocking

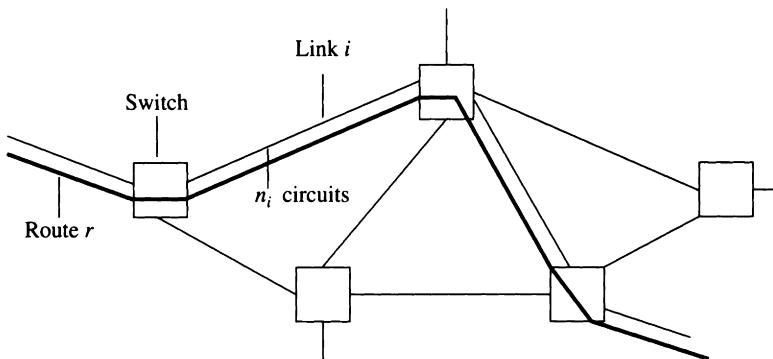
Figure 8.5 illustrates a simple circuit-switched network. The network consists of two groups of telephone sets in two buildings. The telephone sets in each building are connected to a switch (a PBX, or private branch exchange) and the two PBXs are connected by two lines.

You see that it is possible for a call to be requested between the two buildings when the two lines between these buildings are busy. Using a smaller number of lines between the buildings than might be needed in the worst case leads to the possibility of calls being blocked. Of course, in any realistic situation it would be impractical to use the maximum possible number of lines. For



**FIGURE**  
8.5

A small telephone network that consists of two switches and a few telephone sets. It may happen that all the lines between the switches are busy when a user attempts to place a new call. When this occurs, the new call is blocked.



8.6

FIGURE

Circuit-switched network. Switches are connected by links. Link  $i$  consists of  $n_i$  circuits. A route  $r$  is a set of links that form a path.

instance, if the two buildings had 100 telephone sets each, then 100 lines would be needed between the two buildings to allow all the telephone sets to be used at the same time for calls between the buildings. In practice, the likelihood of such a situation is exceedingly small, so that one may reasonably design the telephone system with a smaller number of lines.

The problem faced by the network designer is to select the number of lines between the buildings so that the blocking probability is small enough. We sketch the main features of the methods that the network designer uses. We present the underlying mathematical derivations in the next chapter.

Consider the network of Figure 8.6, which summarizes the basic model that network engineers use to analyze routing in circuit-switched networks. Switches are connected by groups of circuits called links (or trunks). For instance, link  $i$  is composed of  $n_i$  circuits. A circuit provides the bandwidth to transmit one phone call. A route  $r$  is a path in the network from one user to another user. A call in progress along a route uses exactly one circuit on each link along the route. There are  $R$  routes.

As customers place telephone calls, the network uses some routing algorithm to select the routes for the calls. Consequently, calls are placed along the various routes at random times. These calls have variable durations. For instance, assume that the network routes  $N_r$  calls along a particular route  $r$  every minute, on average. Assume also that the average duration of a call is  $T$  minutes. With these assumptions, we would expect that  $v_r := N_r \times T$  calls are in progress at a typical time. Indeed, in  $T$  minutes, about  $v_r$  calls are placed along route  $r$ , and these calls then terminate and are replaced by other calls. Because

of the variability in the times when calls are placed and in the call durations, the actual number  $X_r$  of calls along route  $r$  may be somewhat larger or smaller than  $v_r$ . Consider now a particular link  $i$  with its  $n_i$  circuits. The number  $Y_i$  of calls carried by link  $i$  is the total number of calls along routes that go through link  $i$ . Denote by  $R_i$  the set of routes  $r$  that go through link  $i$ . Then,

$$Y_i = \sum_{r \in R_i} X_r.$$

We expect  $Y_i$  to be approximately equal to  $\sum_{r \in R_i} v_r$ , but we know that  $Y_i$  has some probability of being larger than that value. When  $Y_i$  exceeds  $n_i$  for some link  $i$  along route  $r$ , the call is blocked.

This discussion shows that the probability that a call placed along route  $r$  will be blocked because all the circuits of one of the links along the route are busy is some function of the rates  $\{v_1, \dots, v_R\}$ . We denote that function by  $B_r(v_1, \dots, v_R)$ . We explain algorithms for calculating the function  $B_r$  in Chapter 9.

### 8.2.2 Routing Optimization

We now formalize the best call-routing decision as a solution to an optimization problem. We are given the network topology, that is, the number of circuits  $n_i$  of each link  $i$  between a pair of switches. We are also given the rate  $\lambda_{AB}$  of call requests (per unit of time) between every pair  $(A, B)$  of customer locations.

The routing problem is to select a route for each call so as to maximize the network revenues. We will explain several formulations of the routing problem depending on the information available and the computational effort one is willing to spend.

#### *Static Routing*

One possible formulation of the routing problem is to decide that a call between users at A and B is routed along route  $r_1$  with probability  $p(AB, r_1)$ , along route  $r_2$  with probability  $p(AB, r_2)$ , and so on. Thus, a route  $r$  is selected with probability  $p(AB, r)$  for a call from A to B, and the call is routed along that route if there is a free path along that route. Otherwise, the call is blocked. With this specific routing assignment, we can calculate the rate  $v_r$  of call requests along every route  $r$  from the rates  $\lambda_{AB}$ . We can then calculate the rate of network revenues

$$W := \sum_r w_r v_r [1 - B_r(v_1, \dots, v_R)],$$

where  $w_r$  is the cost rate for calls along route  $r$  and  $v_r[1 - B_r]$  is the rate of calls that route  $r$  actually carries because they are not blocked.

Thus, given the static routing decisions—the probabilities  $p(AB, r)$ , for every pair  $AB$  and route  $r$ —we can calculate the network revenues  $W$ . A good static routing algorithm must come up with routing decisions that yield large network revenues. One method that can be used to develop a good routing algorithm is to design an improvement step that specifies how the probabilities  $p(AB, r)$  can be modified to increase  $W$ . If the improvement step is well designed, then its successive application should lead to routing probabilities that result in a large  $W$ .

Improvement steps based on properties of the gradient of  $W$  with respect to the routing probabilities have been proposed in the literature.

### *Dynamic Alternate Routing*

In static routing, a call is blocked if there is no idle path along the assigned route, even if there is an idle path along a different route. The dynamic alternate routing algorithm takes the network congestion into account in deciding how a call should be routed.

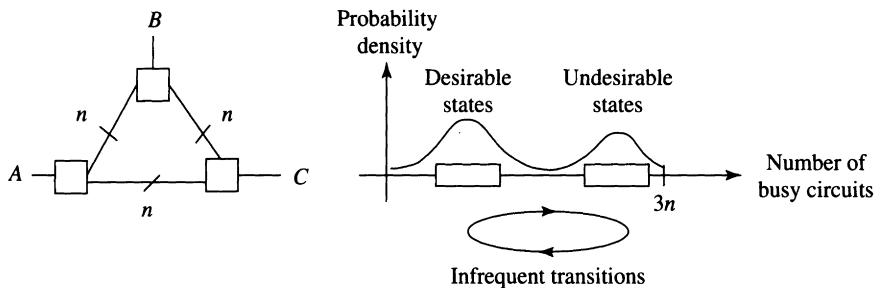
In its basic form, the alternate routing algorithm uses a table that proposes two routes,  $r_1(A, B)$  and  $r_2(A, B)$ , for every pair  $(A, B)$  of customer locations. For instance, these could be the shortest two routes between the locations. An incoming call of type  $(A, B)$  is first assigned route  $r_1$ . If there is no idle path along this route, the call is assigned route  $r_2$ . If there is no idle path along  $r_2$ , the call is blocked.

The advantage of this algorithm over the static routing algorithm is that route selection depends on network congestion. (Of course, in order to implement this algorithm, the state of network congestion must be known.) One disadvantage of the basic dynamic alternate routing algorithm is metastability, as we explain next.

### *Metastability and Trunk Reservations*

*Metastability* is the property of a set of states of a dynamic system that can persist for a long time. When a circuit-switched network uses dynamic alternate routing, a set of states corresponding to a congested network can be metastable. This possibility is illustrated in Figure 8.7.

Consider the network shown in the left part of the figure. It is a symmetric network with three user locations,  $A$ ,  $B$ ,  $C$ , and three switches connected by links with  $n$  circuits each. Assume that the call requests are also symmetric. Specifically, assume that calls between any pair of locations are made with



8.7  
FIGURE

Illustration of metastability. The network on the left is symmetric and calls are placed with rate  $\lambda$  from each node. Each call is equally likely to be for each of the two other nodes. If the network routes a call whenever it finds either a direct path or an indirect (alternate) path, then the probability density of the number of busy circuits is as shown in the figure on the right. This figure illustrates the persistence (metastability) of undesirable network states.

rate  $\lambda$ . The network uses the alternate routing algorithm. For every pair the preferred route  $r_1$  is the direct route through one link, and the alternate route  $r_2$  is the indirect route through two links.

Assuming that the call durations are independent and exponentially distributed, the vector of number of circuits busy in the three links is a Markov chain. (See Chapter 9 for the theory of Markov chains.) We can calculate the steady-state probabilities of that Markov chain by solving the balance equations. The result of this calculation is shown in the right part of the figure. The plot shows the steady-state probability distribution of the total number of busy circuits. This total number can range from 0 to  $3n$ . The plot shows two groups of values with a large probability: one group of small values and one group of large values. The analysis (not shown here) of the time evolution of the network reveals that the number of busy circuits can remain large for a long time before it gets smaller. That is, the state of the network can remain for a long time in a set of undesirable states where many circuits are busy. This set of states is almost stable and is said to be metastable.

We can explain the metastability of congested states as follows. Assume that the network is very congested and that a new call is requested. Since the network is congested, it is unlikely that the new call can be routed along the shortest route. As a result, it is likely that this new call will require two circuits and thereby increase the congestion even further.

Trunk reservation provides an effective protection against the metastability of undesirable congested states. When the routing algorithm uses trunk

reservations, a number of circuits are reserved for routing calls along shortest routes. Let us describe how trunk reservation would be implemented in the network of the figure. First one chooses a number  $m$  smaller than  $n$ . When a new call between  $A$  and  $B$  is requested, the routing algorithm first tries to route the call along the shortest route, using the direct link. If this is not possible, then the algorithm attempts to route the call along the indirect route. That route will be used as long as there are at least  $m$  circuits between  $A$  and  $C$  and between  $C$  and  $B$  that are either used by shortest-route calls or that are free. Thus,  $m$  circuits of each link are reserved for routing calls along their shortest route. This reservation avoids the network reaching a state where most calls are routed along indirect routes, thereby using the network links inefficiently.

See the references cited in section 8.6 for a more detailed discussion of routing in circuit-switched networks.

### *Separable Routing*

Separable routing is also a dynamic routing algorithm, since the route for a new call is chosen on the basis of the network congestion.

The separable routing algorithm calculates its routing decisions by performing one policy improvement step starting with the best static routing algorithm. To understand how this step is performed, consider the following situation. Say that the network is initially in state  $N$ . Here  $N$  is a vector that specifies how many calls are being carried along every possible route in the network. Denote by  $V(N, t)$  the expected network revenues between time 0 and time  $t$  when the best static routing algorithm is used. Now assume that a new call is requested and that two routes, say 1 and 2, can be used for this call. If route 1 is used, then the state will jump from  $N$  to some other value  $N'$  to indicate that there is a new call being carried along route 1. If route 2 is used, the state jumps from  $N$  to  $N''$ . Which is the better decision? The policy improvement step decides that route 1 is preferable if  $V(N', t) > V(N'', t)$  for all large  $t$  and that route 2 is preferable otherwise. Note that the policy improvement step assumes that the static routing algorithm is used at all subsequent times in order to compare the initial decisions.

The crucial question is how we can compare  $V(N', t)$  and  $V(N'', t)$ . That is, we have to evaluate the impact of an additional call on the network revenues. This additional call has a finite holding time. During its holding time, the additional call increases the likelihood that subsequent calls are blocked, which reduces the total network revenues. The effect of one call on the blocking of other calls can be evaluated.

### *Admission Control*

So far we have considered the routing decision, assuming a call is accepted. That is, we have shown how the blocking probabilities of the calls depend on the routing decisions. How should the network decide which calls should be accepted? The future networks will carry many different types of calls: audio, voice, data, video, and so on. These calls differ in terms of their resource needs, revenue generated, request rate, and call duration. Consequently, a good admission policy must base its admission decision on the set of calls currently carried by the network and on the type of call being requested.

We examine a simple model of this admission control problem in section 8.4. We also present a more complex model in Chapter 9.

---

## 8.3 DATAGRAM NETWORKS

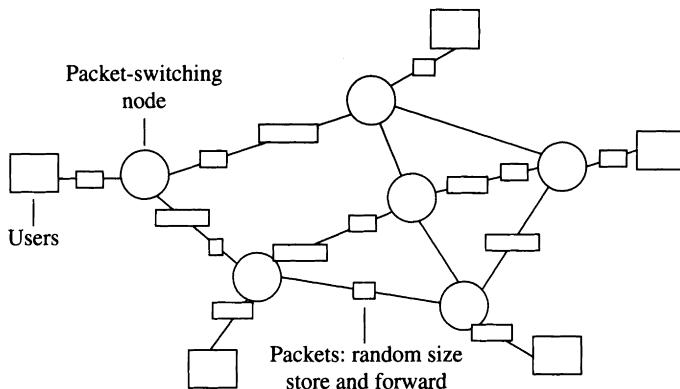
---

When a network transports data as datagrams, it first decomposes the data into packets of variable size. The packets are then sent one by one from node to node along a path to the packet destination. Each packet contains a special field that specifies its destination address. Thus, in a datagram network, the nodes store and forward the individual packets, one at a time. The routes taken by successive packets may be different, even if they go from the same source to the same destination. Also, because the packet sizes may be different, as suggested in Figure 8.8, the transmission times of the packets are different, since they are equal to the packet lengths divided by the transmission rate. In our study of datagram networks we begin by describing a model that we will use to formulate the control questions.

### 8.3.1 Queuing Model

The queuing model of Figure 8.9 can be used by the designer to predict the transmission delays and to design good routing and flow-control algorithms. The top part of the figure shows one node with its components: a receiver converts the optical signal on the incoming fiber into packets. The packets are stored into memory and are then retransmitted on one outgoing fiber.

The bottom part of the figure shows an abstract representation of the same node. The packets are viewed as “customers” or “jobs,” in the language of queuing theory, arriving at random times into a queue. The customers wait for their turn to be served, and the service times are random. The service time is the length of the packet in bits divided by the transmission bit rate. The



8.8

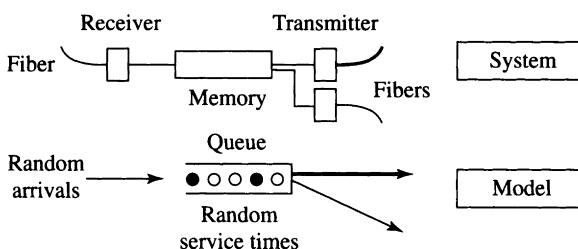
FIGURE

Datagram networks. Packets of different length are transported in a store-and-forward manner.

service time is random when the packet length is random. If the packets all have the same size, as in ATM, the service time is deterministic. Thus, the fluctuations of the arrival times and of the transmission times are modeled by random variables. The specific assumptions about the distributions of these random variables depend on the precise model being used.

### 8.3.2 Key Queuing Result

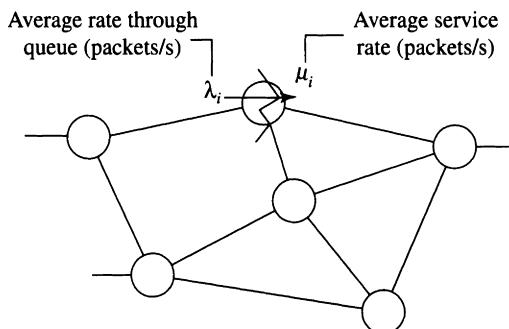
The most useful result of queuing theory for the analysis of datagram networks concerns the network shown in Figure 8.10. That result is the formula for the average delay per packet in such a network.



8.9

FIGURE

Packet-switching node and queuing model.



**8.10**  
**FIGURE**

Queuing network. The symbol  $\lambda_i$  indicates the average rate of packets going through node  $i$  (in packets per second) while  $\mu_i$  is the average transmission rate of that node, also in packets per second, when the node is nonempty.

The network consists of a collection of nodes connected by links. The packets that arrive from outside of the network are assumed to form Poisson processes, and the packet transmission times in the various nodes are assumed to be independent and exponentially distributed. We use the following notation. At each node  $j$ , packets arrive at rate  $\lambda_j$  packets/s, and the service rate is  $\mu_j$  packets/s. Thus  $\mu_j$  is the transmission rate in bits per second divided by the average packet size in bits. It is assumed that  $\mu_j > \lambda_j$  for all  $j$ . Then the average delay faced by a packet entering the network is

$$T = \frac{1}{\gamma} \sum_j \frac{\lambda_j}{\mu_j - \lambda_j}, \quad (8.1)$$

where  $\gamma$  is the total rate at which packets arrive into the network.

The assumptions are not exactly satisfied in actual networks. For instance, the packet lengths are not exponentially distributed. Typically, the packets that travel on the Internet tend to have a bimodal distribution: most packets are either short or long, and the fractions of short and long packets are not consistent with an exponential distribution. Also, since the length of a packet does not change as the packet travels through the network, the transmission times of a packet at the different nodes are not independent. In fact, if one knows the transmission time of a packet at one node, then one can determine the length of that packet and therefore its transmission times in all the other nodes.

Although these assumptions are not always valid, the formula for the average delay per packet provides a reasonably good estimate of the actual

value of that average delay in a real network. This simple formula is the starting point for the construction of routing and flow-control algorithms. (The formula is usually conservative, i.e., the actual delay is smaller than that predicted by the formula.) We derive the delay formula (8.1) in section 9.3.1.

### 8.3.3 Routing Optimization

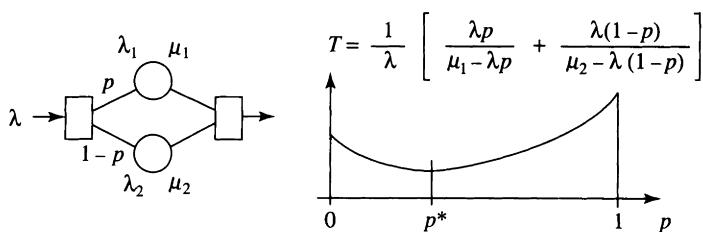
We explain how the delay formula is used to design good routing strategies. We first consider static routing.

#### Static Routing

Consider the simple network illustrated in Figure 8.11. Packets arrive with rate  $\lambda$ . The network can send these packets along two different routes to their common destination. Each route consists of one node, that is, of one buffer equipped with a transmitter. Our objective is to choose the fraction  $p$  of packets that should be sent along the first route so as to minimize the average delay per packet in the network.

We assume that the packets arrive as a Poisson process and that their lengths are independent and exponentially distributed. The two links use transmitters with different rates. Consequently, the two nodes 1 and 2 are modeled as queues with exponential service times with different rates.

To use formula (8.1), we need to identify the parameters in that formula. The parameter  $\gamma$  is the total rate of packet arrivals into the network. That rate is  $\lambda$ . Thus,  $\gamma = \lambda$ . The delay formula contains a sum over all the network nodes. For each node  $j$ ,  $\lambda_j$  designates the average rate of packets going through



8.11

FIGURE

Static routing optimization. Packets are sent to node 1 with probability  $p$  independently of one another and to node 2 otherwise. The right part of the figure shows the average delay  $T$  per packet through the network as a function of  $p$ .

that node, and  $\mu_j$  is the average service rate of that node. Here,  $\lambda_1 = \lambda p$  and  $\lambda_2 = \lambda(1 - p)$ . The service rate  $\mu_1$  of the first node, in packets per second, is equal to the rate of the transmitter in that node, in bits per second, divided by the average length of a packet, in bits. The rate  $\mu_2$  is obtained in a similar manner. Substituting these values in the delay formula gives

$$T = \frac{1}{\lambda} \left[ \frac{\lambda p}{\mu_1 - \lambda p} + \frac{\lambda(1 - p)}{\mu_2 - \lambda(1 - p)} \right].$$

The right part of Figure 8.11 is a plot of the average delay  $T$  as a function of  $p$  for typical values of  $\lambda$ ,  $\mu_1$ , and  $\mu_2$ . That plot shows that the average delay per packet  $T$  is minimized for some value  $p^*$  of  $p$ . Thus, there is an optimal way of splitting the traffic between the two routes.

We now consider the general case. The optimal static routing problem for a general datagram network is to minimize the average delay per packet  $T$  with respect to all routing probabilities.

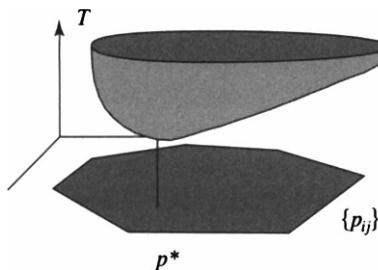
The network sends a packet leaving node  $i$  to node  $j$  with probability  $p_{ij}$ . Given these routing probabilities  $p_{ij}$ , we can calculate the average rates of flow  $\lambda_i$  through the nodes  $i$  by solving the following flow-conservation equations:

$$\lambda_i = \gamma_i + \sum_j \lambda_j p_{ji}, \text{ for all } i.$$

In these equations,  $\gamma_i$  denotes the rate of arrivals of packets from outside the network into node  $i$ . The equations say that, for each  $i$ , the rate of flow  $\lambda_i$  through node  $i$  is equal to the external arrival rate into that node  $\gamma_i$  plus the sum over all nodes  $j$  of the fraction  $p_{ji}$  of the rate  $\lambda_j$  of flow leaving that node  $j$  and being sent to node  $i$ . If the network is open, that is, if all the packets that enter the network can eventually leave it, then the flow-conservation equations have a unique solution  $\{\lambda_i, i = 1, \dots, J\}$ .

Thus, given the external rates  $\{\gamma_i, i = 1, \dots, J\}$ , the flow-conservation equations enable us to calculate the rates  $\{\lambda_i, i = 1, \dots, J\}$  as a function of the routing probabilities  $\{p_{ij}\}$ . Once these rates are determined, we can use our formula (8.1) to compute the average delay  $T$ . Figure 8.12 sketches the delay  $T$  as a function of the routing probabilities  $p_{ij}$ .

The delay  $T$  is a complicated function of the routing probabilities  $p_{ij}$ . The minimization of  $T$  does not result in a closed-form expression for the optimum routing probabilities. Instead, one must use a numerical minimization algorithm. For instance, a gradient projection algorithm can be used to obtain the optimal routing probabilities.



8.12

FIGURE

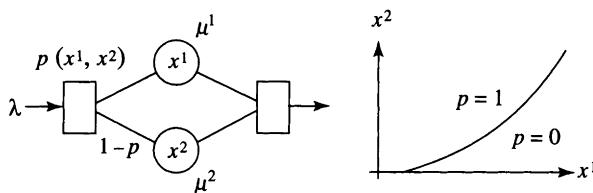
Average delay  $T$  per packet through the network as a function of the routing probabilities  $p_{ij}$ .

### *Dynamic Routing*

Instead of selecting (fixed) probabilities to route the packets when they leave the nodes, we can devise a dynamic routing algorithm that bases the routing decisions on the actual backlogs of the nodes. We illustrate such an algorithm for the simple network of Figure 8.13. The left part of the figure shows a network in which packets arriving as a Poisson process with rate  $\lambda$  can be sent along one of two routes. Each of the two routes is modeled as a single node with exponential service times.

We assume that the routing probability  $p$  can be based on the queue lengths  $x^1$  and  $x^2$ . That is, when a packet arrives, the routing controller looks at the two queue lengths  $x^1$  and  $x^2$  and sends the packet along route 1 with probability  $p(x^1, x^2)$  and along route 2 otherwise.

The designer of the routing algorithm must find the function  $p(x^1, x^2)$  that, when used by the routing algorithm, minimizes the average delay per packet



8.13

FIGURE

Dynamic routing. When the queue lengths are  $x^1$  and  $x^2$ , an arriving packet is sent to queue 1 with probability  $p(x^1, x^2)$ . The graph in the right-hand part of the figure shows the function  $p(., .)$  that minimizes the average delay per packet through the network.

in the network. The solution, that is, the best function, is illustrated in the right part of Figure 8.13. This function specifies that when  $x^1$  is large and  $x^2$  is small, the packets should be sent along route 2, and vice versa. Moreover, if a packet should be sent along route 2 when the queue lengths are  $x^1$  and  $x^2$ , then the same routing decision should be taken when  $x^1$  is larger or when  $x^2$  is smaller.

The proof of these intuitively obvious structural properties of the function  $p(x^1, x^2)$  turns out to be rather involved, even for such a simple network. In fact, very few structural results of this type are known for more complicated networks. Moreover, such structural results do not appear to yield improved procedures for calculating the optimal dynamic routing algorithm.

We now turn to the case of a general network. The derivation of the optimal dynamic routing algorithm for a network with many nodes is a formidable problem that is still beyond the reach of current approaches. Consequently, approximations are necessary. Moreover, the state of the network is not known instantaneously, so that, even if it could be derived, the optimum dynamic routing algorithm would not be implementable. As a result of these limitations, network engineers have developed simple heuristics that can be implemented. Two such heuristics are the Bellman-Ford algorithm and the distributed-gradient algorithm.

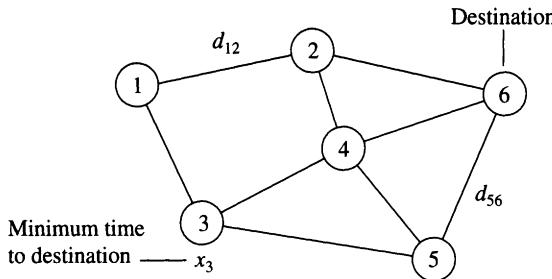
**Bellman-Ford Algorithm** Figure 8.14 summarizes the setup of the Bellman-Ford algorithm. The model is a network of nodes connected by links. The average delay on each link is estimated by the corresponding transmitter. One possible estimation method is for the transmitter on each link to keep track of the backlog in its buffer and to calculate the average delay by dividing the total number of bits stored in the buffer by the transmission rate. The propagation time of signals along the link can be added for improving the estimate.

To explain the calculations performed by the Bellman-Ford algorithm, let us assume that the delay  $d_{ij}$  on the link from node  $i$  to node  $j$  has been estimated for all pairs of nodes. Let us denote by  $x_i$  the minimum delay between node  $i$  and some fixed destination. The minimum delay  $x_i$  must satisfy the equation

$$x_i = \min_j \{d_{ij} + x_j\}. \quad (8.2)$$

These equations are of the form  $x = F(x)$ , where  $x$  designates the vector with components  $x_i$ . Thus, the vector  $x$  satisfies fixed-point equations. These fixed-point equations can be solved by the recursion

$$x^{n+1} = F(x^n).$$



8.14

FIGURE

Setup of the Bellman-Ford algorithm. The problem is to find the shortest path from each node to the destination. It is assumed that each node knows the lengths of the links to which it is attached. Here,  $d_{ij}$  is the length of the link from node  $i$  to node  $j$ , and it is assumed to be known by node  $i$ .

It can be shown that this recursion will converge to the vector of minimum delays for any nonnegative initial vector  $x^0$ . The resulting algorithm is called the Bellman-Ford algorithm.

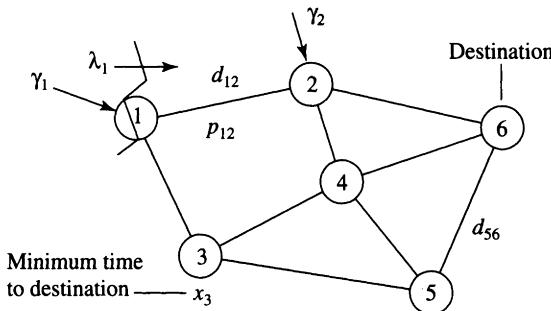
Once the minimum delays have been calculated, the fastest path to the destination is easily identified: a packet that leaves node  $i$  should be sent to node  $j^*$  if  $j^*$  is the value of  $j$  that achieves the minimum in equation (8.2).

**Distributed-Gradient Algorithm** The delays through the links of a network depend on the traffic along those links. Consequently, a routing algorithm can be improved by taking into account the effect of routing decisions on the traffic and therefore on the delays. The distributed-gradient algorithm estimates that effect.

The situation is illustrated in Figure 8.15. Denote by  $\gamma_i$  the rate of traffic entering the network via node  $i$  and by  $\lambda_i$  the average rate of flow through node  $i$ . Also, let  $d_{ij}$  represent the delay along link  $ij$  and  $x_i$  the average delay from node  $i$  to a specified destination. By  $p_{ij}$  we designate the fraction of traffic leaving node  $i$  that is sent to node  $j$ . The problem is to determine the values of these routing probabilities that minimize the average delays from the nodes to the destination. In this algorithm, in contrast to the Bellman-Ford algorithm, the link delays are assumed to depend on the rates of traffic.

The key idea of the algorithm is expressed in the following formula:

$$\frac{\partial x_i}{\partial p_{ij}} = \lambda_i \left[ \frac{\partial d_{ij}}{\partial \lambda_{ij}} + \frac{\partial x_j}{\partial \gamma_j} \right], \quad (8.3)$$



8.15  
FIGURE

Setup of the distributed-gradient algorithm. The problem is to find the routing probabilities that minimize the average delay from each node to the destination. Packets arrive from outside of the network into node  $i$  with rate  $\gamma_i$ . The average delay from node  $i$  to node  $j$ ,  $d_{ij}$  is a function of the rate of packets through that link.

where  $\lambda_{ij}$  designates the rate of traffic through link  $ij$ . This formula calculates the derivative of the delay between node  $i$  and the destination with respect to  $p_{ij}$ . This formula may be understood by multiplying both of its sides by  $\epsilon$ . When  $p_{ij}$  is increased by a small value  $\epsilon$ , the rate through the link  $ij$  increases by  $\lambda_i \epsilon$ . This rate increase has two effects. First, it increases the delay along link  $ij$  by  $\lambda_i \epsilon$  multiplied by the derivative of  $d_{ij}$  with respect to the traffic rate along the link  $ij$ . Second, it increases the traffic rate through node  $j$ . The increase in the traffic through node  $j$  also increases the delay from node  $j$  to the destination and will therefore increase the delay from node  $i$  to the destination by the same amount. The formula expresses these two effects.

To appreciate how the formula can be used, let us assume that each transmitter can estimate the derivative of the form

$$\frac{\partial d_{ij}}{\partial \lambda_{ij}}.$$

Equation (8.3) then provides a recursive procedure for evaluating the terms  $\partial x_i / \partial \gamma_i$ . Suppose node  $i$  is attached to the destination  $s$  by one link  $is$ . The corresponding equation (8.3) is then simply

$$\frac{\partial x_i}{\partial p_{is}} = \lambda_i \frac{\partial d_{is}}{\partial \lambda_{is}}$$

(since  $x_s = 0$ ,  $x_i = d_{is}$ , because a packet originating at  $s$  and destined for  $s$  encounters no delay), and that formula enables us to calculate

$$\frac{\partial x_i}{\partial \gamma_i} = \frac{1}{\lambda_i} \frac{\partial x_i}{\partial p_{is}}.$$

In this way, the terms on the left-hand side of (8.3) can be evaluated for the nodes that are directly attached to the destination. These values can then be used together with equation (8.3) to evaluate the terms corresponding to nodes that are two links away from the destination, and so on.

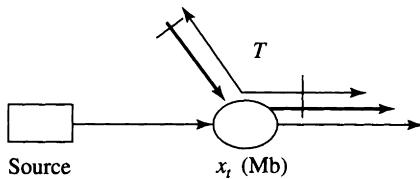
To implement this algorithm, the nodes must broadcast the values of their gradient estimates to their neighbors. Readers should consult the references for details on this algorithm. It should be noted that this algorithm is subject to undesirable oscillations. Remedies for these oscillations have been devised.

### 8.3.4 Congestion Control

*Congestion control* is the name of control procedures that throttle the flow of packets along a path to keep parts of the network from becoming excessively congested.

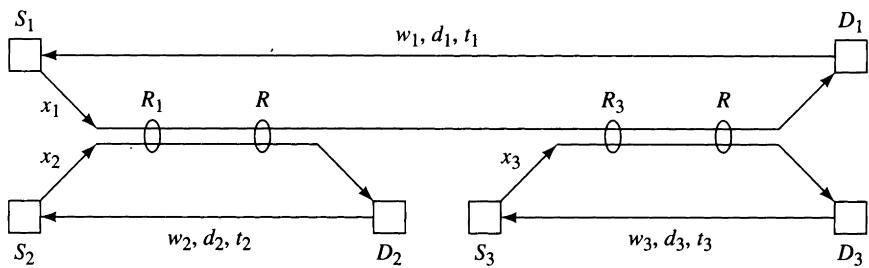
Figure 8.16 illustrates the usefulness of congestion control for an elementary example. Consider a node that is used by two traffic streams represented by the differently shaded arrows. Assume that the traffic flowing on the two top arrows is guaranteed a delay  $T$  less than some specified value  $T_{max}$ .

To meet this guaranteed bound on the delay, the source of the traffic represented by the lower arrows must be prevented from entering the node when more than  $c \times T_{max}$  Mb are already buffered in that node. Here,  $c$  denotes the transmission rate in Mbps.



8.16  
FIGURE

Illustration of congestion control. Two types of packets go through the same node. We assume the packets are served in their order of arrival. If the packets of one type are guaranteed a bounded delay, then the packets of the other type must be stopped when the backlog exceeds a given size.



8.17

FIGURE

Network for study of congestion.

The resulting congestion-control procedure for the traffic flowing along the lower arrows is called a *window congestion control*. A more involved example is explained next.

### **Window Congestion Control**

Figure 8.17 illustrates a set of interconnected routers and hosts. Our objective is to study end-to-end congestion-control mechanisms. Although we make simplifying assumptions about the dynamics of the network, the conclusions of our analysis are relevant to the performance of actual networks.

We assume that the links all have the same rate of 1. Three connections share the network: from source \$S\_i\$ to destination \$D\_i\$, \$i = 1, 2, 3\$. The figure shows transmissions from each destination \$D\_i\$ to the source \$S\_i\$. These transmissions are of acknowledgments. The following quantities are defined for the connection from \$S\_i\$ to \$D\_i\$: the propagation time \$d\_i\$, the transmission rate \$x\_i\$, the round-trip time \$t\_i\$, and the number of bits in transit \$w\_i\$. The routers can store a finite number of bits. If bits arrive at a router that is full, then the bits are dropped. Since these bits do not arrive, they are not acknowledged and the source realizes, after some delay, that the bits were dropped.

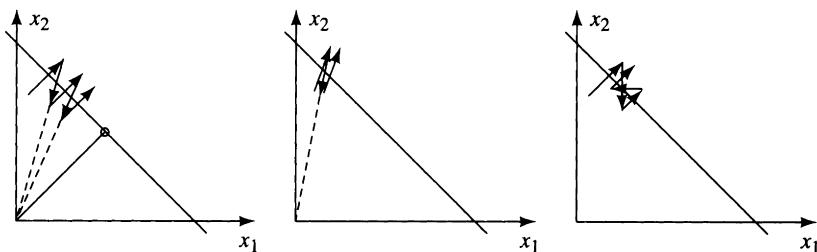
These quantities have the following meaning. Source \$S\_i\$ implements a window congestion control that limits the number of bits in transit to \$w\_i\$. The transmission links that the packets from \$S\_i\$ to \$D\_i\$ and their acknowledgments from \$D\_i\$ to \$S\_i\$ go through have a total propagation time equal to \$d\_i\$. That is, when there is very little traffic in the network, the time between the transmission of a packet and the reception of its acknowledgment by \$S\_i\$ is \$d\_i\$. This delay \$d\_i\$ is called the propagation time of the connection. The round-trip time \$t\_i\$ of the connection is equal to the propagation time \$d\_i\$ plus the queuing time of the packets in the routers. If the links of the network are used efficiently, then \$x\_1 + x\_2 = 1\$ and \$x\_1 + x\_3 = 1\$. Also, if the three connections have a fair share of the links, then \$x\_1 = x\_2 = x\_3 = 0.5\$. The window congestion control algorithm is a

mechanism to adjust the window size  $w_i$  based on the observed quantities  $x_i, t_i$ . We assume that the source gets an opportunity to measure its propagation time  $d_i$  by keeping track of its minimum round-trip time.

The difficulty in designing a window congestion control algorithm is that it must work for a source that does not know the network topology, the link rates, nor the set of other connections with which it is sharing the network.

Before we examine the complete situation of Figure 8.17, let us first assume that the source  $S_3$  is silent. In that case, there are two connections ( $S_1$  to  $D_1$  and  $S_2$  to  $D_2$ ) and only one router ( $R_1$ ) can be a bottleneck. For this simplified situation, a simple algorithm has been implemented. The sources increase their transmission rate until they detect dropped bits. When a source detects that some of its bits were dropped, it reduces its transmission rate by some factor. Figure 8.18 shows the evolution of the rates  $x_1$  and  $x_2$  when the sources implement that algorithm.

The left-hand figure shows the evolution of the rates if we assume that the two sources increase their transmission rate at the same rate and detect the losses at the same time. When the sum of the rates  $x_1 + x_2$  is less than 1, the router  $R_1$  is not congested and it does not drop bits. Accordingly, the sources  $S_1$  and  $S_2$  increase their transmission rate and the vector  $(x_1, x_2)$  moves along a 45-degree line as both components increase at the same rate. Eventually, the sum of the rates  $x_1$  and  $x_2$  exceeds 1. At that time, the buffer of the router fills up and starts dropping packets. Assuming that the two sources discover the losses at the same time, they reduce their transmission rates by 50%. Consequently, the vector  $(x_1, x_2)$  is replaced by  $(x_1/2, x_2/2)$ , as Figure 8.18 shows. The rates then start increasing again, and the vector moves as shown in the left-hand part of the figure. In this ideal case, the vector eventually oscillates around the value  $(1/2, 1/2)$ .



8.18

FIGURE

Two connections that share a single bottleneck. Shown are an ideal case (left), different adjustment rates (center), and different adjustment delays (right).

This algorithm is called *additive increase and multiplicative decrease*. The TCP congestion-avoidance mechanism is of this type. As long as a source does not detect any packet loss, it keeps on increasing its window size linearly. When it detects a loss, the source reduces its window size by a factor 2. The actual mechanism that the sources use to increase their window size results in an increase rate that is inversely proportional to the round-trip time of the connection. Consequently, this algorithm exhibits the behavior sketched in the center part of Figure 8.18. In that part of the figure, one connection has a smaller round-trip time and increases its rate faster than the other connection. The rates eventually oscillate around values such that the connection with the smaller propagation time has a much larger average transmission rate. Moreover, in a real network, the sources learn of packet losses at different times, so that the updates look like those in the right-hand part of the figure. The analysis of this algorithm reveals that even for two connections that share a single bottleneck, the additive increase and multiplicative decrease window congestion algorithm that we just described is biased in favor of the connection with a smaller propagation time.

When there is more than one bottleneck, the situation is even more complex. If we assume that each source  $S_i$  knows  $w_i$ ,  $x_i$ ,  $t_i$ , and  $d_i$ , then an algorithm that converges to the fair and efficient rates has been devised. This algorithm is as follows:

$$\frac{d}{dt} w_i(t) = -\alpha \frac{d_i(w_i - x_i d_i - 1)}{t_i w_i}.$$

In this expression,  $\alpha$  is a parameter that controls the step size of the algorithm. The algorithm converges to a set of connection rates that are a "proportionally fair" equilibrium. By definition, a vector of rates  $(x_1, \dots, x_N)$  is a proportionally fair equilibrium if changing these rates results in a negative sum of their relative increases. That is, if the modified rates are  $(y_1, \dots, y_N)$ , then

$$\sum_{i=1}^N \frac{y_i - x_i}{x_i} \leq 0.$$

To prove the convergence result of the algorithm, one expresses the relations that exist between the quantities  $\{x_i, w_i, t_i, d_i\}$ . One then shows that the algorithm reduces the value of some function of those quantities that is equal to zero only at the limiting point. This function is called a Lyapunov function for the system. (See the references, section 8.6, for details.) In practice, the algorithm explained above is not implemented for two reasons. First, the algo-

rithm assumes that the connections know their round-trip propagation time. Although this time can be estimated, the estimate tends to be noisy, biased against connections that start when the network is already congested, and can be affected by rerouting of connections. Second, the algorithm does not compete well with connections that implement the TCP congestion control.

One simple mechanism has been designed to correct the bias of the TCP congestion control. This mechanism is called *random early drop*, or RED, routers. A router that implements RED drops incoming packets with a probability that is a function of the average recent buffer occupancy. If that average value, computed by a low pass filter, is below a low threshold, then the router accepts the packet. If the average is larger than a high threshold, then the router drops the packet. If the average is between these thresholds, then the router drops the packet with a probability that increases linearly with the average. The effect of this mechanism is that sources learn of the congestion before the router buffer is full and so get a chance to slow down before facing multiple successive losses. Moreover, RED is more likely to drop packets from faster connections (since they send more packets). Consequently, faster connections are more likely to slow down than slow connections, which somewhat corrects the TCP bias.

Many variations on RED have been proposed. One of these variations, called *weighted RED*, classifies packets into a number of classes. The router drops an incoming packet as in RED, except that the threshold values that the router uses to compute the drop probability depend on the class of the packet. In another variation, called *flow RED*, the router maintains counters with the number of packets of the different classes and bases the drop decision on the number of packets of the class of the incoming packet. Yet another variation, called *explicit congestion notification*, marks the packets instead of dropping them. The mark is written in the packet header, and the destination host echoes that mark in the acknowledgment of the packet. The source of the packets adjusts its windows based on the marks that it receives in the acknowledgments.

### ***Rate Congestion Control***

Window congestion control is not suitable when the capacity of the fiber is utilized by a small number of sources. This is because by the time the first packet's acknowledgment by the destination reaches the source, a very large number of packets will have been transmitted, assuming the source does not stop in the meantime. Thus, by the time the destination can signal the source

that some congestion is occurring, it is probably too late for the source to slow down its transmissions. (See section 8.7, problem 10.)

In addition, window congestion control necessitates the establishment of two-way connections to transmit acknowledgments, and this complicates the operations of the network.

To avoid these problems, network researchers are proposing the use of rate-based congestion control. Instead of limiting the number of packets in the network sent by each source, a rate-based congestion control limits the average rate at which sources transmit packets. This control is easier to implement, because it only requires each source to monitor its transmission rate.

For the Internet, a simple rate-based congestion control can be implemented on top of UDP to replace the window-based congestion control of TCP. This rate control operates as follows. The receiver sends a message to the source that specifies the rate at which it wants the source to send the packets. The source implements that rate by computing and controlling an interpacket time. The receiver can calculate when it should receive these requested packets. If some packet arrives late, the receiver can send a modified transmission rate to the source, say half the current rate. If the packets arrive normally, then the receiver can request an increased rate, say a linear increase. To protect this mechanism against lost requests, the source should stop sending after some number of packets if it does not hear from the receiver. One advantage of this strategy over TCP is that the burden is on the receiver to calculate the rate updates, thus making the server simpler.

Another implementation of rate-based congestion control, which is recommended for ATM networks, is the *leaky-bucket* controller. This control mechanism regulates the traffic by smoothing out the bursts of packets that the source would otherwise transmit. We explain this mechanism in the next section.

## 8.4

## ATM NETWORKS

In this section we attend to the control of virtual circuit networks in general and of ATM networks in particular.

We first outline the problems of control of virtual circuit networks. We explain that the user must describe its traffic in a way that can be enforced and monitored. Moreover, the traffic description must enable the network to control the traffic efficiently.

We then study results obtained by using deterministic models. Such deterministic models form the justification of most current recommendations

for the control of ATM networks. In our view, this approach is unnecessarily conservative. Its main, and important, merit is that it is simple to implement.

We conclude the chapter by discussing statistical procedures for specifying traffic characteristics and for controlling the network. We believe that by adopting such procedures, the network can derive significantly higher revenues. However, these statistical procedures require further study. Our presentation outlines a few possibilities. We hope that this section will invite network engineers to study such methods further.

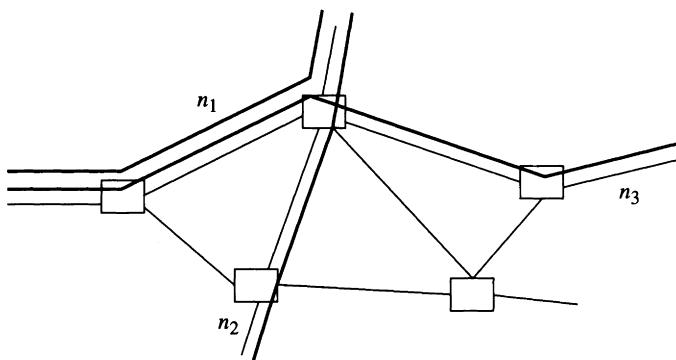
### 8.4.1 Control Problems

We explained in section 8.1 that the user and the network enter into a contract in which the network guarantees a quality of service while the user assures that the traffic will obey specific bounds. Implicit in any contract are provisions for verifying that the terms of the contract are observed by the various parties. Thus, the network must be able to verify that the traffic obeys its descriptors and the user that the quality of service is acceptable. The objective of the control is to maximize the network revenues while meeting the contracted quality of service of the connections. The quality of service specifies the loss rate and delay of the transfer of bits (or cells, or packets).

Other descriptors of the quality of service include security and reliability. Specific temporal characteristics of losses and of delays may be important in particular applications. In multimedia applications, for instance, two users may be connected by a collection of virtual circuit connections. The quality of service might specify bounds on the synchronization offset between the virtual circuit connections. As you see, even the formulation of the problem lends itself to many variations. In this chapter, we limit our attention to loss rate and delay.

We also noted in section 8.1 that a virtual circuit network can exercise four types of control action: admission, routing, flow and congestion control, and allocation of buffers and bandwidth.

To properly design the control procedures of the network, the network engineer must be able to evaluate the performance of specific control procedures. Consider, for example, the network shown in Figure 8.19. The network engineer should be able to determine whether the loss rates and delays of the different virtual circuit connections in this network are acceptable. The answer to that question depends on the characteristics of the traffic generated by the different connections, on the flow-control and allocation strategies, on the buffer capacities in the switches, on the rates of the transmitters, on the



8.19  
FIGURE

Network control. The network engineer needs to determine whether a given set of connections can be carried by the network.

propagation delays and bit error rates of the links, and on the error-correction procedures that the network uses.

The engineer faces three major difficulties in trying to determine the acceptability of given connections. First, the users are often interested in very small loss rates and relatively hard bounds on delays or delay jitter. These loss rates and delay bounds are sensitive to the details of the traffic and are difficult to calculate even for a single node with precise models of the traffic. Second, the network operator never knows precisely the characteristics of the traffic of connections. For instance, even the *mean* rates of compressed video streams vary substantially across movies. Third, the characteristics of the traffic of the virtual circuit connections are modified when they go through nodes and interact with other connections.

One approach to these complex questions is to play it safe and to adopt conservative procedures. This is the approach followed to date by the ATM Forum, as we explain in the next section. These procedures are based on deterministic bounds on the traffic transmitted by the user. The user can enforce these deterministic bounds by using a mechanism called the *leaky bucket*, which we also describe in the next section. The network can verify that the bounds are met by using the same mechanism. Moreover, the network can guarantee deterministic delay bounds and also that no loss occurs when buffers overflow. These procedures based on deterministic bounds may lead the network to carry only a subset of the calls it could carry with statistical procedures.

Deterministic procedures are based on a worst-case analysis, and they do not take into account the likelihood of such worst-case occurrences. Such procedures cannot use the fact that many connections are very unlikely to behave in the worst possible way at the same time. Consequently, deterministic

procedures do not take advantage of the benefits of statistical multiplexing. To take advantage of these benefits, the network must adopt statistical procedures. When using a statistical procedure, the user specifies statistical descriptors of the traffic.

Three questions arise with such a statistical approach. The first question is how the user can enforce statistical characteristics. The second question is how the network can verify that the traffic satisfies these statistical characteristics. The third question is how the network can use the statistics to decide which calls to accept and which to reject.

We explain later practical statistical methods for specifying and verifying traffic characteristics and sketch procedures for controlling the network.

### 8.4.2 Deterministic Approaches

In this section we review the ATM Forum recommendations and network control procedures based on these recommendations.

#### *ATM Forum Recommendations*

The ATM Forum recommendations for the user-network interface specify a mechanism for describing the traffic flowing through a virtual circuit connection. This mechanism, called the *generalized cell rate algorithm* (GCRA), defines the five service categories specified by the ATM Forum: constant bit rate (CBR), variable bit rate (VBR) either real time or non-real time, available bit rate (ABR), and unspecified bit rate (UBR) as we explain below.

The GCRA has two parameters,  $T$  and  $\tau$ , and it times the arrivals of cells as follows. The algorithm defines a *theoretical arrival time*,  $tat$ , of a cell. If the next cell arrives before  $tat - \tau$ , then the algorithm  $GCRA(T, \tau)$  declares that cell to be *nonconformant*, and  $tat$  is unchanged. If it arrives at time  $t \geq tat - \tau$ , then the cell is *conformant* and the algorithm resets the value of  $tat$  to  $\max\{t, tat\} + T$ .

For instance, consider  $GCRA(10, 3)$  with the initial value of  $tat = 0$  and assume that the arrival times of cells are 1, 6, 8, 19, 27. The first cell is conformant since  $1 \geq 0 - 3$ . The algorithm updates  $tat$  to  $\max\{1, 0\} + 10 = 11$ . The second cell is nonconformant since  $6 < 11 - 3$ . The third cell is conformant since  $8 \geq 11 - 3$ , and the algorithm sets  $tat = \max\{8, 11\} + 10 = 21$ . The fourth cell is conformant since  $19 \geq 21 - 3$ , and the algorithm sets  $tat = \max\{19, 21\} + 10 = 31$ . The fifth cell is nonconformant since  $27 < 31 - 3$ . Summarizing, the decisions of the algorithm are C, NC, C, C, NC, where C means conformant and NC nonconformant.

This algorithm is equivalent to a leaky bucket, which we describe next and which we will use to study admission control. Fluid accumulates at a given

rate in a bucket that can store up to  $T + \tau$  units of fluid. Fluid that arrives when the bucket is full is lost. A cell that arrives when the bucket contains less than  $T$  units of fluid is nonconformant. A cell that arrives when the bucket contains at least  $T$  units of fluid is conformant, and it removes  $T$  units of fluid from the bucket. Let us denote by  $F(t-)$  the amount of fluid in the bucket just before time  $t$  and by  $F(t+)$  the amount of fluid just after time  $t$ . The leaky-bucket algorithm is such that a cell is nonconformant if  $F(t-) < T$  and then  $F(t+) = F(t-)$ . Otherwise, the cell is conformant and  $F(t+) = F(t-) - T$ . If the next cell arrives  $s$  time units later, then  $F(t+s-) = \min\{F(t+) + s, T + \tau\}$ .

In our previous example, assume that the bucket contains  $T + \tau = 13$  units of fluid at time 0. Then,  $F(1-) = 13$ ,  $F(1+) = 3$ ,  $F(6-) = 8 = F(6+)$ ,  $F(8-) = 10$ ,  $F(8+) = 0$ ,  $F(19-) = 11$ ,  $F(19+) = 1$ ,  $F(27-) = 9 = F(27+)$ . Consequently, the successive decisions are  $C, NC, C, C, NC$ .

The ATM Forum recommends that *constant bit rate* (CBR) traffic on a line with rate  $R$  should specify its peak cell rate ( $PCR$ ) and its cell delay variation tolerance ( $CDVT$ ). The meaning of these parameters is that the cells should be conformant for the

$$\text{GCRA} \left( \frac{R}{PCR}, CDVT \right)$$

algorithm.

Thus, if  $R/PCR = 5$  and  $CDVT = 0$ , then the peak cell rate is 20% of the line rate, and the cells arrive at multiples of 5 cell transmission times. If  $R/PCR = 5$  and  $CDVT = 1$ , then the fastest arrival times are  $-1, 4, 9, 14, 19$ , and so on. This sequence corresponds to a periodic stream with rate equal to 20% of the line rate: one arrival every fifth cell transmission time. If  $R/PCR = 4.5$  and  $CDVT = 1$ , then cells can arrive at times  $-1, 4, 8, 13, 17, 22, 26, 31$ , and so on. This sequence has rate  $R/4.5$ . An easy way to see why the maximum rate of a CBR stream is indeed PCR is to recall that a cell takes away  $T = R/PCR$  from the leaky bucket, which is filled at a specified rate. Thus, over a long time interval with duration  $t$ ,  $t$  units of fluid enter the bucket and at most  $t/T$  cells can arrive. The maximum rate is therefore  $1/T = PCR/R$  cells per cell transmission time  $1/R$  second, or  $PCR$  cells per second.

The intuitive meaning of  $\text{GCRA}(R/PCR, CDVT)$  is that the cells can arrive at their peak cell rate  $PCR$  but do not have to be exactly periodic. The  $CDVT$  measures the departure from exact periodicity. Such departure may be necessary because of the framing structure that carries the cells. For instance, imagine a CBR stream that corresponds to 3.5 cells every frame time of a physical layer. In one implementation, the successive frames might carry 3 or 4 cells, and such framing introduces a cell delay variation. Similarly, multiplexing and the

injection of maintenance cells introduce a cell delay variation. Note that a late cell delays the set of all future acceptable arrival times. Thus, it is not correct to think of a CBR stream as having arrival times being multiples of  $R/PCR$  with a shift of  $CDVT$ .

For *variable bit rate* (VBR) traffic, the ATM Forum specifies the  $PCR$ ,  $CDVT$ , burst tolerance ( $BT$ ), and the sustained cell rate ( $SCR$ ). The meaning of these parameters is that the cells should be conformant for both

$$\text{GCRA} \left( \frac{R}{PCR}, CDVT \right)$$

and

$$\text{GCRA} \left( \frac{R}{SCR}, BT + CDVT \right)$$

algorithms.

The motivation for this definition of VBR is that such a stream might be very bursty and send a number of back-to-back cells separated by idle periods. The  $BT$  parameter makes such bursts acceptable.

For instance, consider a VBR stream with  $R/PCR = 1$ ,  $R/SCR = 20$ ,  $CDVT = 0$ ,  $BT = 57$ . If we think back about the leaky-bucket interpretation of the  $\text{GCRA}(R/SCR, BT + CDVT) = \text{GCRA}(20, 57)$  algorithm, we see that a conformant cell takes away 20 units of fluid. With an initial content of  $T + \tau = 77$  units of fluid, we find that four cells can arrive back to back, at times 0, 1, 2, 3. Indeed,  $F(0-) = 77$ ,  $F(0+) = 57$ ,  $F(1-) = 58$ ,  $F(1+) = 38$ ,  $F(2-) = 39$ ,  $F(2+) = 19$ ,  $F(3-) = 20$ ,  $F(3+) = 0$ . A new group of four cells can then arrive at time 80 because  $F(80-) = 77$ . Thus, bursts of four cells can arrive at times 0, 80, 160, and so on. Such a stream is also conformant for the  $\text{GCRA}(R/PCR, CDVT) = \text{GCRA}(1, 0)$  since this controller makes all the streams conformant. The bursty stream that we constructed has a long-term average rate equal to  $4/80 = 1/20$ , since it has bursts of size 4 every 80 time units. Thus, the sustained cell rate of the stream  $SCR$  is indeed such that  $R/SCR = 20$ .

As another example, consider the parameters  $R/PCR = 5$ ,  $R/SCR = 10$ ,  $CDVT = 16$ ,  $BT = 20$ . This stream can have bursts of five consecutive cells with bursts at time 0, 1, 2, 3, 4, 50, 51, 52, 53, 54, 100, 101, 102, 103, 104, 150, 151, and so on. The sustained cell rate is  $1/10$  of the line rate (five cells every 50 cell transmission times).

The ATM Forum is currently developing specifications for *available bit rate* (ABR) transmissions. The operating principle is that an ABR connection may have a guaranteed minimum cell rate and an imposed peak cell rate. The

actual rate available to the connection varies between these two bounds on the basis of feedback information about the congestion in the network and destination. The rate adjustment scheme is a time-varying GCRA control that operates end to end and is rate-based. The ATM Forum does not specify the rate-control algorithm that end systems and switches must use but defines general mechanisms. The ATM cells contain a flow-control field that the switches fill to indicate the congestion level they experience. Also, the sources and destinations can send resource management (RM) cells.

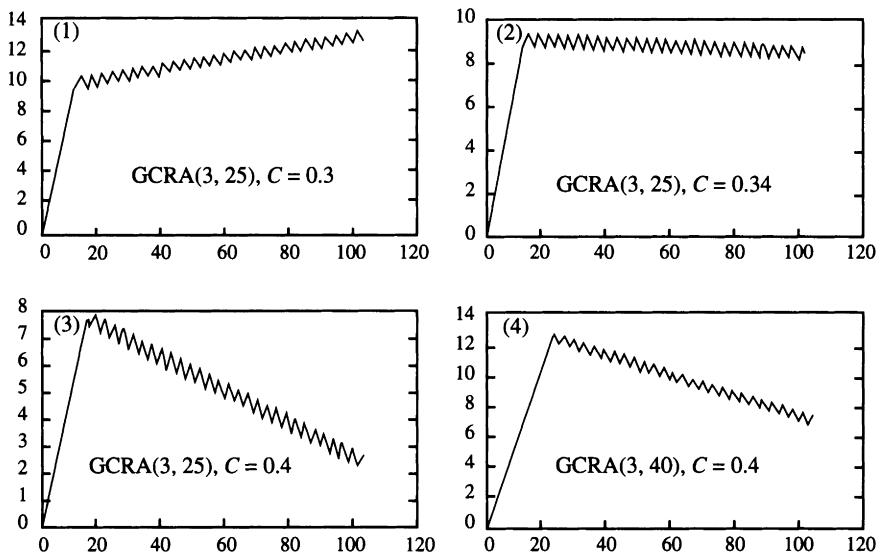
In a typical scheme, the sources periodically send RM cells to indicate their desired rate. The switches then compare the requests of various VCs and distribute their spare bandwidth accordingly. The switches indicate their bandwidth allocation in returning RM cells that go back to the source. In addition, the switches set congestion indication bits in the forward cells and in the returning RM cells. The destination monitors the congestion indication bits of the arriving cells and marks the corresponding bit of the RM cells accordingly. The source uses the congestion indication bits of the RM cells and the bandwidth allocation values of these cells to modify the parameters of its GCRA. Many variations are possible and are being explored. For more information, the reader should keep track of the revised versions of the ATM Forum recommendations. Thus, ABR is an attempt to use feedback to better utilize the network.

The last service being specified by the ATM Forum (in addition to CBR, VBR, and ABR) is *unspecified bit rate* (UBR). UBR is best-effort service with no guarantee on the quality of service.

### ***Admission Control***

How many  $\text{GCRA}(T, \tau)$  connections can go through a buffer equipped with a transmitter that sends  $C$  cells per unit of time if the delay through the buffer must be less than  $D$  units of time? Here, one unit of time is a cell transmission time on each one of the incoming connections.

To answer this question we first study the following problem. We want to find the *fastest* sequence of cells that is conformant to the  $\text{GCRA}(T, \tau)$ . We say that a sequence of arrival times  $\{x_1, x_2, \dots\}$  is faster than another sequence of arrival times  $\{y_1, y_2, \dots\}$  if  $x_k \leq y_k$  for all  $k \geq 1$ . We construct this fastest sequence by filling up the leaky bucket with  $T + \tau$  units of fluid at time 0 and by sending a cell and removing  $T$  units of fluid as soon as the bucket contains  $T$  units of fluid.



8.20

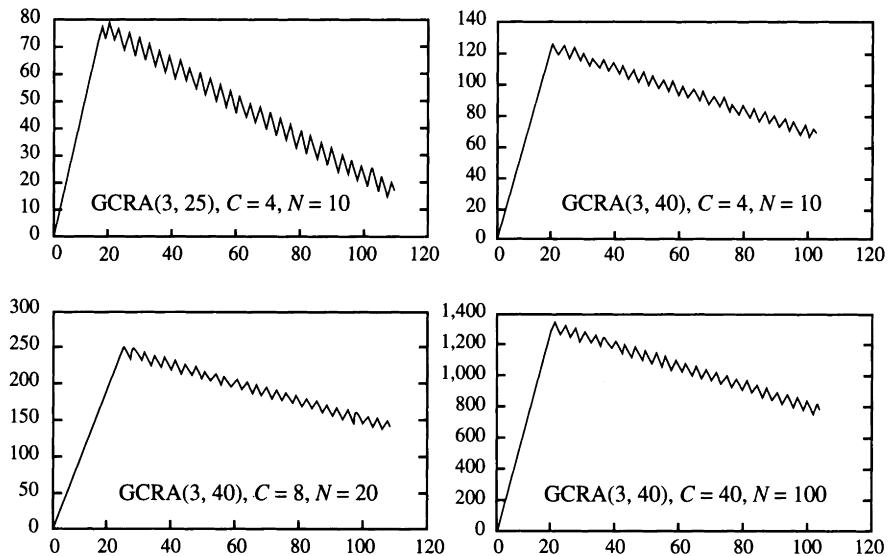
FIGURE

Buffer occupancy for the fastest streams with the parameters indicated. In (1), the system is unstable because  $C$  is too small. The other cases show the effect of the burstiness of the stream on the backlog.

Assume now that a  $\text{GCRA}(T, \tau)$  stream goes through a buffer equipped with a transmitter that transmits  $C$  cells per unit of time. We want to analyze the maximum backlog of the buffer, assuming that it is initially empty. The maximum buffer backlog will occur if the cell stream is the fastest. If  $C > 1/T$ , then the buffer is faster than the arrival stream (since  $1/T$  is the maximum long-term rate of a  $\text{GCRA}(T, \tau)$  stream) and the maximum backlog is finite.

Figure 8.20 shows the evolution of the buffer occupancy for the fastest streams possible with given GCRA parameters. Case (1) is unstable. This instability is caused by  $C$  being smaller than  $1/T$ . Case (2) is stable since  $C > 1/T$ . By comparing cases (3) and (4) we observe the effect of the burstiness of the stream.

If  $N$  streams that conform to  $\text{GCRA}(T, \tau)$  share the same buffer, then the maximum occupancy of the buffer occurs again when the streams are the fastest. We present a few examples in Figure 8.21. By comparing these cases we observe that multiplying the number of sources  $N$  and the service rate  $C$  by the same factor  $V$  results in multiplying the worst-case occupancy by  $V$ .



8.21  
FIGURE

Buffer occupancy when  $N$  fastest streams possible with  $\text{GCRA}(T, \tau)$  share a buffer with service rate  $C$ .

This should be expected since both the number of arrivals and the number of departures in each time step are multiplied by  $V$ .

A good approximation of the maximum backlog in the buffer with rate  $C$  and  $N$   $\text{GCRA}(T, \tau)$  streams can be derived as follows. Assume that  $\tau \gg T$ . Then  $K + 1$  cells can be sent back-to-back by a single stream if

$$\tau + T - KT + K \geq T.$$

Indeed, during the time interval  $[0, K]$ ,  $K$  units of fluid enter the leaky bucket and  $KT$  units of fluid are removed by the first  $K$  cells. The left-hand side of the above inequality is the amount of fluid that is left at time  $K$ , and this amount should be larger than  $T$  if the  $(K + 1)$ th cell is conformant. The maximum number of back-to-back cells is therefore approximately equal to

$$M := \frac{\tau}{T - 1} + 1 = \frac{\tau + T - 1}{T - 1}. \quad (8.4)$$

After these back-to-back cells, the other cells follow each other approximately every  $T$  time units, the time required to collect enough fluid in the leaky bucket to send a new cell. Our description neglects round-off effects that are not significant for our analysis. If each of  $N$  streams produces these  $M$  back-

to-back cells during the interval  $[0, M - 1]$ , then the buffer accumulates  $N \times M$  cells and can serve only  $(M - 1) \times C$  cells, so that the backlog is approximately  $B(T, \tau, C, N)$ , where

$$B(T, \tau, C, N) := N \times M - (M - 1) \times C \approx N \frac{\tau + T - 1}{T - 1} - \frac{C\tau}{T - 1}.$$

If  $T \times C > N$ , as must be assumed for stability, then the cells stop accumulating after the burst of back-to-back cells, so that  $B$  is the maximum backlog.

For the numerical examples of Figures 8.20 and 8.21 this formula gives

$$B(3, 23, 0.34, 1) = 8.59, B(3, 25, 0.4, 1) = 8.5, B(3, 40, 0.4, 1) = 13$$

$$B(3, 25, 4, 10) = 85, B(3, 40, 4, 10) = 130$$

$$B(3, 40, 8, 20) = 260, B(3, 40, 40, 100) = 1300.$$

Comparing these numbers with the figures shows that this approximation is satisfactory.

Let us go back to the question that we asked at the beginning of this section. Assume that the maximum acceptable delay through the buffer is  $D$  cell transmission times. We want to estimate the maximum number  $N$  of  $\text{GCRA}(T, \tau)$  streams that the network can accept.

To answer the question, we observe that the maximum backlog must be at most  $D \times C$  cells. Indeed, a cell that faces a backlog of  $D \times C$  experiences a delay equal to  $D$ . Consequently, the maximum number  $N$  is obtained by solving

$$B(T, \tau, C, N) = D \times C,$$

that is,

$$N \times M - (M - 1) \times C \approx N \frac{\tau + T - 1}{T - 1} - \frac{C\tau}{T - 1} = D \times C.$$

By solving this equation we find

$$N \approx C \frac{D(T - 1) + \tau}{T + \tau - 1}.$$

Recall that these derivations assume that  $C > N/T$ . An equivalent way to look at the above equation is to write the constraint on  $N$  as

$$N \times \alpha_G(T, \tau, D) \leq C, \quad (8.5)$$

where

$$\alpha_G(T, \tau, D) := \max \left\{ \frac{T + \tau - 1}{D(T - 1) + \tau}, \frac{1}{T} \right\}. \quad (8.6)$$

We can interpret the inequality (8.5) as stating that each GCRA( $T, \tau$ ) connection with maximum delay  $D$  requires a bandwidth  $\alpha_G(T, \tau, D)$  given by (8.6). We can call  $\alpha_G(T, \tau, D)$  the *effective bandwidth* of a GCRA( $T, \tau$ ) connection with maximum delay  $D$ .

If we recall the formula (8.4) for the maximum number  $M$  of back-to-back cells, we find that

$$\alpha_G(T, \tau, D) := \max \left\{ \frac{M}{D + M - 1}, \frac{1}{T} \right\}. \quad (8.7)$$

Thus, if  $D$  is small compared with  $M$ , the effective bandwidth is close to 1. That is, the switch must treat such a source as a constant bit rate source with a rate equal to the line rate  $R$ . At the other extreme, if  $D$  is much larger than  $M$ , then the effective bandwidth is close to  $M/D$  and about  $D/M$  can be accommodated (so long as  $T > D/M$ ).

For instance, one finds that

$$\alpha_G(3, 40, 30) = 0.42, \alpha_G(3, 40, 20) = 0.52, \alpha_G(3, 40, 10) = 0.70.$$

Using the ATM Forum parameters for a VBR connection, we see that

$$\alpha_G(R/SCR, BT + CDVT, D) = \max \left\{ \frac{\beta + 1}{\beta + D}, \lambda \right\}, \quad (8.8)$$

where we introduce the parameters

$$\lambda := \frac{SCR}{R} \text{ and } \beta := \frac{BT + CDVT}{R/SCR - 1}. \quad (8.9)$$

This effective bandwidth is a measure of the cost of carrying such a VBR connection. Note that the cost increases as the acceptable delay decreases. The cost also increases with the burstiness measured by  $\beta$ .

The delay constraint becomes binding if  $D$  is small enough or if  $\beta$  is large enough. Otherwise, the bandwidth is determined by the average rate  $\lambda$ .

Summarizing this section, we have explored the implications of the GCRA control mechanism recommended by the ATM Forum. We have analyzed the maximum number of connections that can go through a buffer subject to a maximum delay constraint. The analysis is based on the worst-case behavior of the connections and ignores the likelihood of such behavior. The result can

be viewed as requiring an effective bandwidth per connection. The effective bandwidth increases with the burstiness of the connection and decreases with the acceptable delay.

### Pricing Calls

We explore the pricing implications of deterministic approaches in a simple model that highlights some features of the problem.

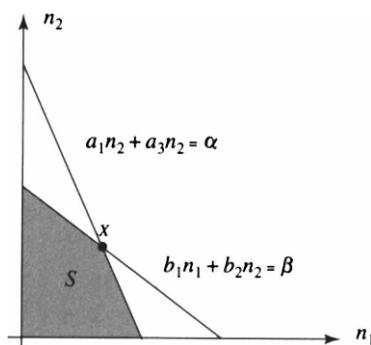
Consider calls of two types: 1 and 2. Each call requires resources of two classes,  $A$  (e.g., buffer space) and  $B$  (e.g., bandwidth), in order to be carried by the network. Assume that a call of type  $i$  requires  $a_i$  units of resources of class  $A$  and  $b_i$  units of class  $B$  (for  $i = 1, 2$ ). The total amount of resources of classes  $A$  and  $B$  available are  $\alpha$  and  $\beta$ , respectively.

Denote by  $n_i$  the number of calls of type  $i$  carried by the network for  $i = 1, 2$ . The set of admissible pairs  $(n_1, n_2)$  is

$$S := \{(n_1, n_2) \in \{0, 1, 2, \dots\}^2 \mid a_1 n_1 + a_2 n_2 \leq \alpha \text{ and } b_1 n_1 + b_2 n_2 \leq \beta\}.$$

This set is illustrated in Figure 8.22. Assume for the time being that the parameters of the problem are such that the two oblique boundary lines cross each other at some point  $X$ , as shown in the figure.

We consider that the network will charge  $p_i$  for a call of type  $i$  ( $i = 1, 2$ ) per unit of time, and we examine the unit prices  $(p_1, p_2)$  that the network can charge in a competitive environment. To simplify the problem, we assume that



8.22

**FIGURE**

Set  $S$  of acceptable numbers of calls of two types that require resources of two classes. We show that when the constraint lines intersect as here at a point  $X$ , a competitive network should operate at the intersection point.

the demand is larger than the supply, so that we can attract as many customers as we can carry, if our price for the service is competitive. Let us call the network N.

We define the competitive environment as follows. Assume that N offers the prices  $(p_1, p_2)$  and that some other network N' offers the prices  $(p'_1, p'_2)$ . (N' has the same resources,  $\alpha$  and  $\beta$ .) If the revenues per unit of time R of N and  $R'$  of N' are such that  $R' \geq R$ , then it must be that  $p_i \leq p'_i$  if N carries calls of type  $i$ , for  $i = 1, 2$ . The interpretation of this condition is that if N' is as profitable as N, to stay in business N cannot charge more than N'; otherwise, N would lose its customers.

Network N wants to find the prices it can charge to collect revenues at rate R. (R may equal the cost of the resources plus return on investment.)

Our first observation is that if the network chooses to carry only calls of type 1, then it can obtain the rate of revenues R by charging  $p_1 = Ra_1/\alpha$  since the network can carry  $\alpha/a_1$  calls of type 1 (see figure). Consequently, it must be that

$$p_1 \leq Ra_1/\alpha.$$

Indeed, if N charges  $(p_1, p_2)$  with  $p_1 > Ra_1/\alpha$ , then it cannot compete with a network N' that charges  $(Ra_1/\alpha, \infty)$  and carries only customers of type 1.

Similarly, we find that

$$p_2 \leq Rb_2/\beta.$$

Some algebra shows that there is a pair of prices  $(p_1^*, p_2^*)$  that satisfies the above two conditions with strict inequalities that yields the revenues R when the network operates at the operating point X of the figure. Moreover, when N offers these prices, the only networks that can be competitive with N must offer the same prices and operate at the same point X.

Thus, when the boundary lines cross, it is optimal (necessary) for the network to carry both types of calls. To make the discussion more concrete, consider that calls of type 1 are CBR with rate  $\lambda$  and that calls of type 2 are VBR with GCRA( $T, \tau$ ) where  $\tau \gg T$ .

Imagine that  $n_1$  calls of type 1 and  $n_2$  calls of type 2 go through a buffer with capacity B and with transmitter rate C. No losses are acceptable. The first condition required is

$$n_1\lambda + n_2\frac{1}{T} \leq C,$$

so that the transmitter can keep up with the average rate of the calls.

The second condition concerns the buffering. In the worst case, calls of type 2 send  $M$  back-to-back cells followed by periodic cells every  $T$  time units, where

$$M = \frac{\tau + T - 1}{T - 1},$$

as we explained in the previous section. The maximum backlog in the buffer occurs at the end of the  $M$  back-to-back cells and is equal to  $M(n_1\lambda + n_2 - C)$ . Thus, the second condition is

$$n_1M\lambda + n_2M \leq B + CM.$$

The two constraint lines cross each other if and only if

$$B < CM(T - 1) = C(\tau + T - 1).$$

### 8.4.3 Statistical Procedures

In this section we describe statistical procedures for specifying traffic and quality of service and for controlling the network.

#### *Traffic Models*

Network engineers use a few different stochastic models for specifying traffic in high-performance networks. Stochastic fluid models view traffic as a fluid with a randomly varying rate. Queuing models view packets or cells as discrete entities that arrive one at a time or in finite batches.

#### *Markov-Modulated Fluids*

A Markov-modulated fluid (MMF) is a nonnegative function of a finite continuous-time Markov chain. Accordingly, an MMF is given as  $r(z_t)$ , where  $\{z_t, t \geq 0\}$  is a continuous-time Markov chain on some finite state space  $Z$  with rate matrix  $Q = \{q(i, j), i, j \in Z\}$  and  $r : Z \rightarrow [0, \infty)$  is a given function. ( $r$  may be measured in bits or cells per second.)

The interpretation of the definition is that  $r(z_t)$  is the rate at time  $t$  of some bit stream. The rate fluctuates randomly. This model is motivated by observing the bit stream that a variable bit rate video compression algorithm generates.

Assume that the MMF  $r(z_t)$  goes through a buffer equipped with a transmitter with rate  $c$ . We want to analyze the evolution in time of the buffer occupancy process  $x_t$ . The analysis enables us to determine how large the buffer capacity should be and what delays the MMF faces through the buffer.

We show in section 9.3.4 that the buffer occupancy process has an invariant distribution of the form

$$P(x_t > x) = \sum_{z \in \mathbb{Z}} a(z) e^{-\beta_z x}, \quad x \geq 0.$$

In this expression,  $\{\beta_z, z \in \mathbb{Z}\}$  are the eigenvalues of the matrix  $A = [A(i, j), i, j \in \mathbb{Z}]$ , where

$$A(i, j) = q(i, j)/(r(j) - c).$$

To determine the coefficients  $a(z)$ , one must calculate all the eigenvalues and eigenvectors.

To simplify the analysis, one usually approximates this sum of exponentials by a single exponential,

$$P(x_t > x) \approx K e^{-\beta x}, \quad \text{for } x \gg 1, \quad (8.10)$$

where  $\beta$  is the smallest of the eigenvalues  $\{\beta_z, z \in \mathbb{Z}\}$  and  $K$  is the corresponding coefficient. The calculation of  $K$  requires computing all the eigenvalues and eigenvectors.

A useful example is when  $r(z(t))$  is the sum of the rates of  $N$  independent and identically distributed sources, each modeled by on-off Markov fluid. That is, each source is modeled by a Markov chain on  $\{0, 1\}$  with rate matrix such that  $q(0, 1) = \lambda > 0$  and  $q(1, 0) = \mu > 0$ . When the source is in state 1, it produces fluid at rate 1. When it is in state 0, the source does not produce fluid. Assume that the superposition of these  $N$  sources goes through a buffer with transmission rate  $C$ . Then one finds

$$\beta = \frac{1 + \lambda - N\lambda/C}{1 - C/N} \quad (8.11)$$

and

$$K = \left( \frac{N\lambda}{C(1 + \lambda)} \right)^N \prod_{z=1}^{N - \lfloor c \rfloor - 1} \frac{\beta_z}{\beta_z - \beta}, \quad (8.12)$$

where  $\lfloor c \rfloor$  is the integer part of  $C$ .

To estimate  $K$  one must solve for all the eigenvalues  $\beta_z$ , which can be done numerically easily as long as  $N$  is not too large.

This approach has the merit of yielding complete results for simple models. However, the analysis of networks with Markov-modulated fluid sources using this approach appears beyond reach.

In the following section we discuss a different approach.

### More General Model

The Markov-modulated traffic model describes the random fluctuations of the rate of a bit stream. A more general traffic model is a random process  $A := \{A(t), t \geq 0\}$  that specifies the number  $A(t)$  of bits carried by the traffic during the interval  $[0, t]$  for  $t \geq 0$ . Without further assumptions, this model is too general for us to be able to analyze how the network can transport such traffic. Somehow we must specify the average rate of the traffic and some measure of burstiness.

The average rate is  $[A(t + T) - A(t)]/T$  for large  $T$ . Under suitable assumptions, this average rate is a well-defined quantity  $\lambda$ . That is,  $[A(t + T) - A(t)]/T$  approaches  $\lambda$  as  $T$  increases, and that value  $\lambda$  does not depend on the realization of the stochastic process  $A$  nor on  $t$ . The process  $A$  has these properties if it is stationary and ergodic.

### Averaging Rate Fluctuations

The rate of a traffic stream fluctuates. For instance, the peak rate of a video stream may be 10 times larger than its average rate. To prevent losses, a network node could allocate to each stream a bandwidth equal to the peak rate of that stream. However, such an allocation is overly conservative, and it is sometimes possible for the transmitter to allocate a bandwidth much closer to the average rate than to the peak rate. There are two fundamentally different methods that a network can use to reduce the bandwidth it must allocate to each connection. The first method is multiplexing many sources. The other method is buffering.

The multiplexing method exploits the fact that different sources fluctuate independently so that when the rate of a source is larger than average, the rate of another may be smaller than average. Consequently, the rate of the superposition of many sources tends to be close to its average value. This observation is similar to the fact that if one throws 1,000 fair coins, about 500 of them land on heads. The probability that more than 600 coins will land on heads is very small, about  $10^{-10}$ . Thus, if each of 1,000 sources is off 50% of the time and transmits at rate  $\alpha$  the other 50% of the time, then the total rate of the sources rarely exceeds  $600 \times \alpha$ . The analysis of the multiplexing method determines how many sources must be multiplexed and the transmission rate per source required so that the probability that the total rate of the sources exceeds the transmitter rate is smaller than some specified value, say  $10^{-10}$ .

The buffering method uses the fact that over a long duration  $[t, t + T]$  a stream  $A$  with rate  $\lambda$  produces a total number of cells close to  $\lambda T$ . Consequently, if  $c > \lambda$ , then  $A(t + T) - A(t) \leq cT$  with a large probability. Now, if it were true that  $A(t + T) - A(t) < cT$  for all  $t \geq 0$ , then a node that transmits the stream

with rate  $c$  would delay it by at most  $T$ . To see this, note that the buffer cannot remain nonempty for  $T$  consecutive time units. Indeed, if the buffer is empty at time  $t$  and nonempty during  $[t, t + T]$ , then during that interval it has output  $cT$  bits and it must be that more than  $cT$  bits entered the buffer, that is,  $A(t + T) - A(t) \geq cT$ , a contradiction. If the buffer cannot remain nonempty for  $T$  consecutive time units, then every bit that enters must leave before  $T$  time units. This argument shows that, by using a buffer, a node can transmit at a rate  $c$  only slightly larger than the average rate  $\lambda$ . The node delays the bit stream by at most  $T$  if  $A(t + T) - A(t) \leq cT$  for all  $t \geq 0$ . In the statistical approach, we do not insist that this inequality hold all the time, but only that it hold with a large probability. We then conclude that the node delays the bits by at most  $T$  with a large probability. The analysis of this method estimates the transmitter rate needed so that the node delays the bit stream by more than  $T$  seconds with some specified small probability, say  $10^{-10}$ .

How do these methods compare? Multiplexing is effective for real-time traffic and buffering is effective for non-real-time traffic. The intuitive justification for this statement is that bursty real-time traffic cannot be buffered long enough to average out its rate fluctuations. For instance, a video connection has a rate close to the peak rate for as long as a few seconds, say 10 s. These long periods of high bit rate correspond to active scenes in the video, say a car chase in an action movie. The motion compensation video compression algorithm is not effective during fast-changing scenes and, consequently, produces a large bit rate to reflect the rapid modifications of the successive frames. If the node buffers the video bit stream for less than a few seconds, then it still must transmit the stream at a rate close to the peak rate. One might think that by superposing a large number, say 100, of such video bit streams, buffering would become much more effective. However, analysis and simulations show that not to be the case. We show below that buffering is ineffective for real-time streams such as video bit streams. However, multiplexing can be effective for such streams. In contrast to the real-time traffic case, buffering is effective for streams that can be delayed by a few seconds in the network, such as streams produced by interactive applications. Obviously, buffering is the way to handle best-effort traffic (UBR or ABR).

### **Multiplexing**

We now present the main results on the multiplexing of many sources. Remember that the objective of multiplexing is to use a transmission rate per source close to the average rate of each source instead of requiring a rate close to the

peak rate. Network engineers call this possibility of reducing the required rate per source the *multiplexing gain*.

Consider  $N$  sources. For  $n = 1, \dots, N$ , let  $Y_n$  be the rate at some time  $t$  of source number  $n$ . We assume that the sources are stationary, independent, and identically distributed. That is, the rates  $\{Y_1, \dots, Y_N\}$  are independent random variables that have a common distribution that does not depend on  $t$ . We want to find the rate  $c$  such that

$$P\{Y_1 + \dots + Y_N > cN\} \leq 10^{-9}.$$

In words, if a node transmits the superposition of the  $N$  sources with that rate  $c$ , then it drops at most a fraction  $10^{-9}$  of the bits.

As we may expect, the rate  $c$  is slightly larger than the average rate, say  $\lambda$ , of each source. The precise value of  $c$  depends on  $N$  and on the distribution of the random variables  $Y_i$ . Using the Bahadur-Rao theorem (see section 9.4.5), we find

$$P(Y_1 + \dots + Y_N > Nc) \approx \frac{1}{\sqrt{2\pi}\sigma\theta_c\sqrt{N}} e^{-NI(c)}. \quad (8.13)$$

In the above expression,  $\theta_c$  achieves the maximum in

$$I(c) = \sup_{\theta} [\theta c - \varphi(\theta)],$$

where

$$\varphi(\theta) = \log E[\exp(\theta Y_1)]$$

and

$$\sigma^2 = \varphi''(\theta_c) = \frac{\partial^2 \varphi}{\partial \theta^2}(\theta_c).$$

(The term  $\varphi(\theta)$  is called the *logarithmic moment generating function*.) In the case of on-off sources with  $P(on) = p$  and peak rate  $a$ , the coefficients of (8.13) have the following form:

$$\theta_c = \frac{1}{a} \log \left( \frac{c(1-p)}{p(a-c)} \right), I(c) = \frac{c}{a} \log \left( \frac{c(1-p)}{p(a-c)} \right) - \log \left( \frac{a(1-p)}{(a-c)} \right), \sigma^2 = c(a-c).$$

As a numerical example, we use the following values for the homogeneous on-off sources:  $\lambda = 1/20s$ ,  $\mu = 1/5s$ ,  $a = 18$  Mbps,  $c = 8$  Mbps. These parameters correspond to  $P(on) = \lambda/(\lambda + \mu) = 0.2 = p$  and therefore to a mean rate equal to  $p \times a = 3.6$  Mbps.

We find that  $\theta_c = 0.0646$ ,  $I(c) = 0.1523$ ,  $\sigma^2 = 80$ . We can then calculate the value of  $N$  needed so that  $P\{Y_1 + \dots + Y_N > cN\} \leq 10^{-9}$ . The computer finds that  $N \geq 118$  is the required condition.

Note that the number of sources that are in the on state has a binomial distribution. Thus the probability that the aggregate input rate exceeds the output rate can be represented exactly as

$$\sum_{k \geq Nc/a}^N \binom{N}{k} p^k (1-p)^{N-k}.$$

We can evaluate this expression directly and find  $P\{Y_1 + \dots + Y_N > cN\} \leq 10^{-9}$  for  $N \geq 114$ . You will note the remarkable accuracy of the Bahadur-Rao approximation. The Bahadur-Rao approximation can be used for complex distributions of the random variables  $Y_k$  where a direct calculation is very complex. Note that for such distributions, the evaluation of the parameters that enter (8.13) must be performed numerically.

The Bahadur-Rao theorem enables us to analyze the case of multirate sources. That theorem can also be used to analyze the overflows of mixtures of different types of sources. To do this, say that we have 50% of sources of type  $Y$  and 50% of sources of type  $Z$ . We can then construct a hybrid source that is of type  $Y + Z$ . The analysis of such a situation reveals that the value of  $c$  required to achieve a small loss probability cannot be written as the sum of the necessary rates for the  $Y$  sources and the  $Z$  sources. Thus, unfortunately, for small buffers there is no additive result similar to the effective bandwidth.

### *On-Line Estimation*

The discussion above shows that the network needs detailed information about the statistics of the sources in order to determine the capacity that it should allocate to connections. In practice it may not be realistic to expect the users to know that information when they set up the connection. These contradicting aspects seem to make statistical procedures impractical. We believe that this conclusion is not correct. The network could guarantee a quality of service by being conservative in its initial admission control and measure the traffic to determine the actual resources that the traffic requires. For instance, the admission control could be based on the peak rate of the traffic. Once the connection was in progress, the network could monitor its actual requirements. (The network also could calculate the price from the actual resources that the connection utilized.) Such a procedure presents a risk that all the ongoing

connections might, after a while, suddenly become much more active and require more bandwidth. However, such an event is very unlikely.

According to the above description, we propose four methods for estimating the actual bandwidth that connections require. The methods differ in their numerical complexity and in their efficiency.

Assume that identical and independent sources with known mean rate  $m$  and peak rate  $a$  want to be transmitted by a transmitter with rate  $C$ . We want to design an on-line admission procedure that accepts the maximum number of sources subject to a loss probability of  $10^{-9}$ . We assume that nothing is known about the sources other than their mean and peak rates.

We first assume that the sources are on-off with  $P(on) = m/a$ . On-off sources are easily seen to be the most bursty sources with given mean and peak rates in that fewer of them can be accepted for a given loss rate. We use the Bahadur-Rao formula to determine the maximum number  $N_0$  of such sources that can be accepted, as we did in the previous section. We then accept the  $N_0$  sources, and we measure their statistics.

The four methods we propose differ in how they infer the actual number of sources that can be carried by the transmitter. In methods 1 and 2, we calculate the bandwidth needed to carry the  $N_0$  sources with a loss rate of  $10^{-9}$ . In methods 3 and 4, we estimate the parameters of the Bahadur-Rao formula, and we calculate the maximum value  $N$  that can be carried.

**Method 1** We divide the interval between the mean rate  $m \times N_0$  of the  $N_0$  sources and  $C$  into a number of equal parts. That is, we calculate  $C_0 = m \times N_0$ ,  $C_1, C_2, \dots, C_K = C$  so that  $C_1 - C_0 = C_2 - C_1 = \dots = C_K - C_{K-1}$ . We then monitor the instantaneous rate of the  $N_0$  ongoing connections and determine the loss rate  $L_k$  that these connections would face if the service rate were  $C_k$  instead of  $C$ , for  $k = 0, 1, \dots, K$ . This monitoring is performed in parallel, by a device that does not perturb the connections. We then find  $C(N_0) = \min\{C_k | L_k \leq 10^{-9}\}$ .

We could decide that  $C(N_0)$  has been determined if its value has stopped fluctuating for some time. More research is required to determine satisfactory stopping rules.

For instance, if  $C = 155$  Mbps and  $C(N_0) = 120$  Mbps, we are led to think that 25% more calls could be accepted. We can then accept a few more calls and repeat the above procedure.

This method has the advantage of being simple to implement.

**Method 2** This second method is a modified version of Method 1 and results in faster estimation. We define  $N_0$  and  $C_k$  as in method 1, and we accept

$N_0$  calls. We group the  $N_0$  calls into two subgroups. Subgroup 1 has 40% of the calls, and subgroup 2 has the remaining 60%. In parallel, we measure for each value of  $k = 0, 1, \dots, K$  the loss rates  $L_k^1$  and  $L_k^2$  that the two groups would face if they were transmitted with respective bandwidth  $0.4C_k$  and  $0.6C_k$ . We use these numbers to estimate the loss rate  $L_k$  that  $N_0$  calls would face with bandwidth  $C$  according to the formula

$$L_k = 1.1619(L_k^1)^{-2}(L_k^2)^3.$$

This formula is derived from the Bahadur-Rao formula (8.13), which shows that the loss rate  $L(N)$  for  $N$  calls has the form

$$L(N) = \frac{A}{\sqrt{N}} \exp\{-NG\}$$

so long as the bandwidth per call  $c$  is constant. By measuring  $L_k^1 = L(0.4N_0)$  and  $L_k^2 = L(0.6N_0)$  with  $c_k = C_k/N_0$ , we can determine the two unknown parameters  $A$  and  $G$  and calculate  $L_k = L(N_0)$ .

The motivation behind this approach is that the subgroups have fewer calls and benefit less from statistical multiplexing. Consequently, loss rates  $L_k^1$  and  $L_k^2$  are substantially larger than  $L_k$  and are faster to estimate.

**Method 3** The third method accepts the same number,  $N_0$ , of calls as the previous two methods and monitors these calls to estimate the coefficients of the Bahadur-Rao formula. The estimation is based on the approximation

$$\varphi(\theta) := \log E[\exp(\theta Y_1)] \approx \frac{1}{N_0} \log \frac{\int_0^T \exp\{\theta X(N_0, t)\} dt}{T}, \text{ for } T \gg 1,$$

where  $X(N_0, t)$  is the total instantaneous rate of the  $N_0$  calls.

This approximation assumes that the processes are ergodic and stationary, so that

$$\frac{\int_0^T \exp\{\theta X(N_0, t)\} dt}{T} \rightarrow E \exp\{\theta X(N_0, t)\} = E \exp\{\theta(Y_1 + \dots + Y_N)\}.$$

Once  $\varphi(\theta)$  has been estimated in parallel for a large number of different values of  $\theta$ , we can determine  $\theta_c$ ,  $I(c)$ , and the other required parameters. We can then find out the value of  $N$  that would result in the desired loss rate.

**Method 4** This final method is similar to method 3 but uses the independence of the calls in a different way. The estimation is based on the approximation

$$\varphi(\theta) := \log E[\exp(\theta Y_1)] \approx \log \frac{\sum_{n=1}^N \int_0^T \exp\{\theta X_n(t)\} dt}{N_0 T}, \text{ for } T \gg 1,$$

where  $X_n(t)$  is the instantaneous rate of the  $n$ th call.

This estimation uses the fact that the random variables

$$\frac{\int_0^T \exp\{\theta X_n(t)\} dt}{T}, n = 1, \dots, N$$

are independent and identically distributed and converge (by ergodicity and stationarity) to  $E \exp\{\theta(Y_1)\}$ .

The method then proceeds as method 3. By better exploiting the independence of the calls, this method obtains estimators with a lower variance.

Our experiments suggest that the second method represents a good trade-off between complexity and speed. The fourth method is the fastest but requires complex computations.

### **Buffering**

In this section, we consider a situation where the loss rate is kept small by using a large buffer. The buffer stores bursts of cells that arrive faster than they can be transmitted. It is unlikely that the bursts are frequent enough to make the buffer overflow.

Except for very simple source models (e.g., Poisson or a Markov-modulated process with a small number of states), it is difficult to analyze exactly the small loss rate at a large buffer. The cause of the difficulty is that the state space of a Markov model of the source and buffer system is large, which makes the numerical analysis complex. Because of that complexity, and with the objective of deriving tractable results, we turn to an asymptotic analysis of the loss rate as the buffer becomes large. Not surprisingly, the loss rate becomes smaller as the buffer increases. When the buffer is large, the loss rate is well approximated by an exponential function of the buffer size, as we already saw in (8.10).

Roughly, the loss rate is approximately  $\exp\{-BI(C)\}$ , where  $B$  is the buffer size (in cells, say) and  $I(C)$  is some increasing function of the transmitter rate  $C$  and, obviously, of the statistics of the traffic. We argue that we should choose  $C$  large enough so that  $\exp\{-BI(C)\}$  is small enough. For a video source, we might want  $\exp\{-BI(C)\} \approx 10^{-10}$  when  $B = 1,000$ . Thus, we want  $I(C) \approx 1\%$ . For a database source, we might want  $\exp\{-BI(C)\} \approx 10^{-8}$  for  $B = 20,000$ , so that  $I(C) \approx 0.1\%$ . We designate this target value of  $I(C)$  by  $\delta$ . Thus,  $\delta = 1\%$  for video and  $\delta = 0.1\%$  for database. (Once again, recall that these are working hypotheses and not standards.)

Now suppose that there are  $J$  types of traffic, and  $n_j$  sources of type  $j$  are multiplexed onto an output link. We want

$$\lim_{B \rightarrow \infty} \frac{1}{B} \log P(W \geq B) \leq -\delta,$$

where  $W$  is the buffer occupancy. Under appropriate assumptions, this constraint can be satisfied when

$$\sum_{j \in J} n_j \alpha_j(\delta) \leq C, \quad (8.14)$$

where  $C$  is the total output link rate and  $\alpha_j(\delta)$  is the effective bandwidth for the type  $j$  source corresponding to  $\delta$ .

Inequality (8.14) allows a simple policy for call acceptance that is analogous to that of the traditional circuit-switched networks, since the effective bandwidth for each call can be determined independently of the other types of calls. Furthermore, since  $\alpha_j(\delta)$  lies between the mean and peak rates of the source, the difference between the peak rate and  $\alpha_j(\delta)$  is the bandwidth saving through multiplexing.

The effective bandwidth  $\alpha(\delta)$  of a source that produces a random number  $A(t)$  of cells in  $t$  seconds can be calculated as

$$\alpha(\delta) = \frac{\Lambda(\delta)}{\delta}, \quad (8.15)$$

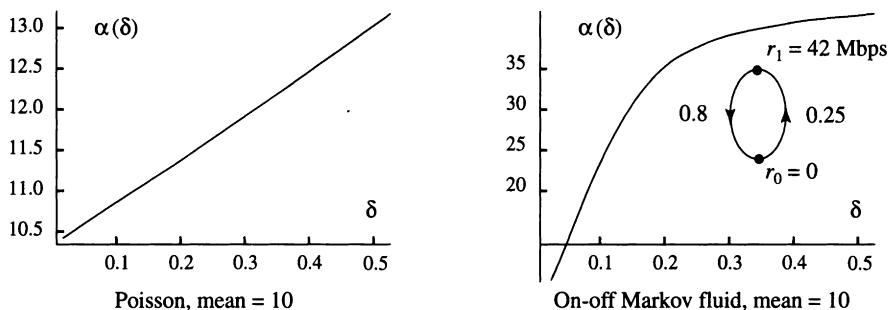
where

$$\Lambda(\delta) = \lim_{t \rightarrow \infty} \frac{1}{t} \log E\{e^{\delta A(t)}\}. \quad (8.16)$$

Figure 8.23 shows the effective bandwidth for two types of sources. The left-hand source produces *independent, identically distributed* (iid) batches of bits that are Poisson distributed with mean rate 10 per unit of time. The right-hand source is an on-off Markov-modulated fluid with mean rate 10 and with the parameters shown in the figure. The on-off source has a larger effective bandwidth than the Poisson source.

When  $\delta$  is small enough, one may be able to justify the following approximation:

$$\log E\{e^{\delta A(t)}\} \approx \log \left[ 1 + \delta E\{A(t)\} + \frac{\delta^2}{2} E\{A(t)^2\} \right] \approx \delta E\{A(t)\} + \frac{\delta^2}{2} E\{A(t)^2\},$$



8.23

FIGURE

Effective bandwidth of iid Poisson and of on-off Markov-modulated sources with the same average rate. Note that these effective bandwidths differ substantially.

which results in

$$\alpha(\delta) \approx \lambda + \frac{1}{2}\delta D^2, \quad (8.17)$$

where

$$\lambda := \lim_{t \rightarrow \infty} \frac{1}{t} E\{A(t)\} \text{ and } D^2 := \lim_{t \rightarrow \infty} \frac{1}{t} E\{A(t)^2\}.$$

In these definitions and in (8.17),  $\lambda$  is the average rate of the stream, and  $D^2$  is called its *dispersion*. This simple approximation for  $\delta \ll 1$  shows that the effective bandwidth increases with the burstiness of the stream and indicates that an approximate measure of burstiness (for large buffers and small  $\delta$ ) is the dispersion. When  $\delta \ll 1$ , we are willing to lose quite a few cells, and the second moments of the stream are good predictors of the loss rate, as one might guess from a functional central limit theorem. When  $\delta$  is larger, the losses are determined by the tail behavior, and the higher moments cannot be neglected in the calculation of the effective bandwidth.

Formulas or algorithms are available for calculating the effective bandwidth of a large class of models. Methods for on-line estimation of the effective bandwidth are the subject of current research and so are adaptive techniques for selecting a suitable value of  $C$ .

These results deal with a single buffer and may be usable for a local ATM network. When traffic goes through multiple buffers, the situation is more complex. When streams share a buffer, they interact and modify one another's statistics and effective bandwidth. At first the problem appears intractable: the statistics of a stream depend on those of all the streams it interacted with, and

the same is true for the latter streams. Fortunately, a simplification occurs. One can show that if the transmitter rate  $C$  of a buffer is large enough, then a stream preserves its effective bandwidth as it goes through the buffer. Specifically, stream  $j$  preserves its effective bandwidth if  $C$  is larger than the sum of  $\alpha_j^*(\delta)$  and the average rate of all the other streams that share the buffer. Here,  $\alpha_j^*(\delta)$  is the *decoupling bandwidth* of stream  $j$ . Formulas for calculating that decoupling bandwidth are given in Chapter 9 where the applications of that result to call admissions are discussed.

The approach above works well only if the buffer is large. Numerical and simulation experiments show that admission control based on the notions of effective and decoupling bandwidth may be too conservative. What is happening is that the method is based on the estimate of the exponential rate of decay of the loss probability, and it ignores the preexponential factor, which may be very small.

### *Statistical Multiplexing and Buffering*

In many networks, a large number of sources are multiplexed and losses are further reduced by buffering. In such networks, the two effects that we discussed in the previous sections are combined.

The analysis of the combined effect of statistical multiplexing and buffering is rather complicated and does not yet carry over to more than a single queue. Nevertheless, the methods help one to understand the behavior of queues.

We present two approaches. The first approach yields satisfactory approximations for small buffers. The second approach is more suitable for larger buffers, but it is not as accurate in estimating the effect of statistical multiplexing.

**Approach 1: Small Buffer** To analyze the loss rate at a buffer of size  $B$  that serves  $N$  sources with rate  $Nc$ , we argue that losses occur when two events happen: first, the aggregate rate of the  $N$  sources must reach the value  $Nc$ ; second, the rate must remain large enough until the buffer overflows. The probability of the first event is given by the Bahadur-Rao formula (8.13). The probability of the second event was obtained by Weiss,

$$P[\text{Buffer overflow} \mid \text{total rate} > Nc] = \exp \left\{ - \left( N \frac{B}{\beta} \right)^{1/2} K(Nc) \right\},$$

where  $\beta$  is the burst size of one source and  $K(Nc)$  is a constant that depends on  $Nc$ . Combining the two results, the probability of overflow can be estimated as

$$\frac{1}{\sqrt{2\pi}\sigma\theta_c\sqrt{N}} \exp \left\{ -NI(c) - \left( N \frac{B}{\beta} \right)^{1/2} K(Nc) \right\}, \quad (8.18)$$

where

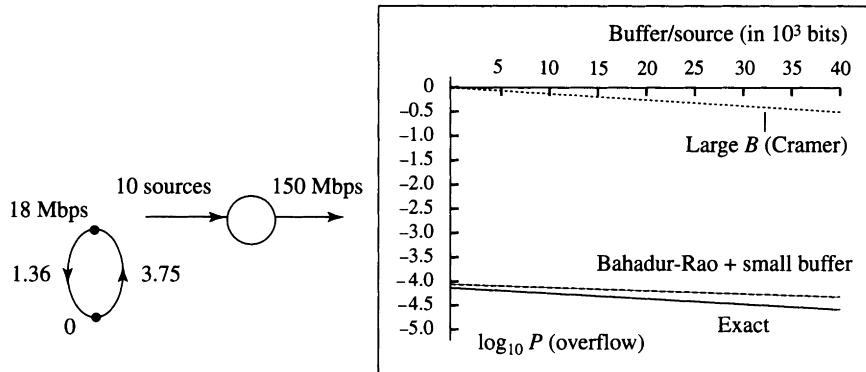
$$\sigma^2 = \frac{M''(\theta_c)}{M(\theta_c)} - c^2, \quad M(\theta) := E[\exp(\theta Y_1)]$$

and  $\theta_c$  achieves the maximum in

$$I(c) = \sup_{\theta} [\theta c - \varphi(\theta)].$$

The burst size of a source is defined as the product of the peak rate of the source times the mean holding time of that peak rate.

Figure 8.24 illustrates this formula. In that figure, 10 sources are served by a buffer with rate 150 Mbps. The sources are modeled by on-off Markov-modulated fluids with the parameters shown in the figure. The graphs show the loss rate as estimated by the small- $B$  asymptotics, by the combination of small- $B$  and Bahadur-Rao estimate, and the exact loss rate that can be calculated for this simple model.



8.24

FIGURE

The figure compares three methods for calculating the loss rate at a small buffer. The first method is based on Cramer's theorem and ignores the statistical multiplexing gain. The second method combines the Bahadur-Rao estimate for 0-buffer and Alan Weiss's analysis of the overflow of small buffers. The third method, possible only for simple systems, is an exact calculation of the loss rate.

**Approach 2: Many Sources** We limit the discussion to homogeneous sources. Let  $N$  be the number of incoming traffic streams. Assume that all sources are independent of one another. In particular, we consider the case that the arrival process from each source is modeled by a Markov-modulated fluid with two states. The off ( $= 0$ ) and on ( $= 1$ ) states have exponentially distributed holding times with parameters  $\lambda$  and  $\mu$ , respectively. The output rate of a source is  $a$  in the on state and 0 in the off state. This on-off Markov model has received much attention in the research community because of its ability to model bursty processes such as sampled voice and its usefulness in queuing analysis.

Assume that the output buffer has first in, first out (FIFO) service discipline. Let  $b$  and  $c$  denote respectively the amounts of buffer space and bandwidth per source. Assume that  $\frac{\lambda}{\lambda+\mu}a < c < a$ , where the first inequality ensures stability and the second inequality allows a nonzero probability for the aggregate input rate to exceed the output rate.

We now explain the approximations to be made. Assume that time is discretized into epochs, with  $X_n$  equal to the number of cell arrivals from a single source in epoch  $n$ . Define

$$\varphi_m(\theta) = \frac{1}{m} \log E \left[ \exp \left( \theta \sum_{n=1}^m X_n \right) \right].$$

Assume that the asymptotic logarithmic moment generating function, defined as

$$\varphi(\theta) = \lim_{m \rightarrow \infty} \varphi_m(\theta),$$

exists. Denote  $\Phi(c, b, N)$  as the proportion of time that the buffer is full.

**Theorem 8.4.1** Under appropriate assumptions,

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \Phi(c, b, N) = -I(c, b),$$

where

$$I(c, b) = \inf_m \sup_\theta [\theta(b + mc) - m\varphi_m(\theta)]. \quad (8.19)$$

**Proof** The intuitive explanation of this result is that if an overflow occurs at time 0, then there must have been some time  $-m$  at which the buffer was last empty and since when at least  $N(b + mc)$  cells have arrived. The probability of this many arrivals decays exponentially with  $N$ . The most likely way for an

overflow to occur corresponds to the duration  $m$  with the smallest exponential decay rate.

### Multiclass Case

We explained in the previous discussions that real-time traffic can be handled without buffering. The network determines the number of connections that it can accept so that the probability is very small that the total rate of the connections exceeds the available bandwidth. The interactive connections are handled with buffering. The network also estimates the bandwidth required per connection so that the probability of buffer overflows is very small. In the following we explain how the network can handle these two traffic types simultaneously. The basic idea is that the nodes give priority to real-time traffic and low priority to interactive traffic. Thus, the real-time traffic is not affected by the interactive traffic. However, the interactive traffic only gets the bandwidth that the real-time traffic does not use. The key question for us is to determine how the network should take that effect into account.

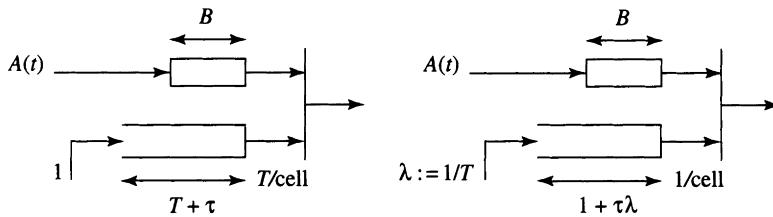
The real-time traffic with instantaneous rate  $\mathbf{v} := \{v(t), t \geq 0\}$  that goes to a transmitter with rate  $C$  occupies a bandwidth equal to the effective bandwidth of  $\mathbf{v}_C := \{\min\{v(t), C\}, t \geq 0\}$ . Thus, if the real time traffic  $\mathbf{v}$  and an interactive traffic  $\mathbf{d} := \{d(t), t \geq 0\}$  go through a transmitter with rate  $C$ , the transmitter gives priority to  $\mathbf{v}$  over  $\mathbf{d}$  and makes sure that

- ◆  $P\{v(t) \geq C\}$  as estimated by Bahadur-Rao is acceptably small;
- ◆  $EB(\mathbf{v}_C) + EB(\mathbf{d}) \leq C$ ;
- ◆ the decoupling conditions are satisfied.

### Choosing a GCRA

Consider a user application that generates a random stream that must be transported using a service characterized by GCRA. What parameters of GCRA should the traffic stream request? In practice, the user will try various combinations among the values made available by the network until the cheapest acceptable parameters are identified. With some experience, equipment and service providers will be able to advise users and to recommend specific parameters for a given application.

In this section we look at the question from a theoretical angle and try to identify the set of parameters that is acceptable to carry a given traffic. The objective of the exercise is to understand the key statistics that affect that selection.



**FIGURE**  
8.25

We use a leaky bucket  $\text{GCRA}(T, \tau)$  to regulate the stream  $A(t)$  in the left-hand figure. The bottom buffer is the fluid buffer that provides the permits to transmit cells. We redefine units in the figure on the right-hand side so that one cell takes away one unit of fluid. Fluid enters at rate  $\lambda = 1/T$ .

Consider using a  $\text{GCRA}(T, \tau)$  controller with  $\tau \gg T$  to shape some process  $\{A(t), t \geq 0\}$ . That leaky bucket is equipped with a cell buffer as shown in the left-hand side of Figure 8.25.

We redefine the units of fluid so that one cell requires one new unit of fluid. With this new unit, the fluid enters at rate  $\lambda = 1/T$ . The modified system is shown in the right-hand part of Figure 8.25.

We want to determine the values of  $B$ ,  $T$ , and  $\tau$  so that the probability that the buffer overflows is some small value, say  $10^{-8}$ . These parameters will also give us a bound on the delay  $B/\lambda = BT$  caused by traffic regulation. Alternatively, we could specify the maximum acceptable delay  $D$ , then calculate  $B = \lambda D = D/T$  and find parameters  $T$  and  $\tau$  that result in an acceptable loss probability.

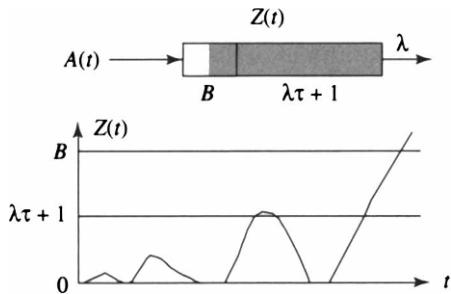
To analyze the loss probability, we note that the system shown in Figure 8.25 is equivalent to the system shown in Figure 8.26. The interpretation of a backlog  $z$  in this system is that if  $z < \lambda\tau + 1$ , then  $\lambda\tau + 1 - z$  is the amount of "token fluid" available in the bottom buffer of the right-hand part of Figure 8.25; if  $z > \lambda\tau + 1$ , then  $z - \lambda\tau - 1$  is the amount of cells backlogged in the top buffers of Figure 8.25.

To analyze the probability of overflowing the cell buffer, we can analyze the probability that  $Z(t)$  exceeds  $B + \lambda\tau + 1$ . If  $\lambda > \alpha(\delta)$ , the effective bandwidth of  $A$ , then this probability is approximately  $\exp\{-\delta(B + \lambda\tau + 1)\}$ . Thus, if we want this probability to be approximately  $10^{-8} \approx e^{-18.3}$ , then we need

$$\delta(B + \lambda\tau + 1) = 18.3$$

and

$$\lambda \geq \alpha \left( \frac{18.3}{B + \lambda\tau + 1} \right),$$



8.26

FIGURE

A backlog less than  $\lambda\tau + 1$  means that there is a supply of tokens available. A backlog larger than  $\lambda\tau + 1$  means that cells are backlogged.

that is,

$$\frac{1}{T} \geq \alpha \left( \frac{18.3}{B + \tau/T + 1} \right).$$

Using  $B = D/T$  we can rewrite the previous inequality as follows:

$$\frac{1}{T} \geq \alpha \left( \frac{18.3}{(D + \tau)/T + 1} \right).$$

This last inequality gives us the trade-off between parameters  $T$  and  $\tau$  of the GCRA( $T, \tau$ ) that we should request from the network. The best choice corresponds to the cheapest tariff. Note that the best choice depends on the statistics of the traffic through its effective bandwidth. Remember that our analysis ignores the factor of the exponential in calculating the loss probability.

### **Traffic Shaping**

The sources can use a simple procedure to reduce the resources that they require from the network. This procedure is called *traffic shaping*. We explain how the source can implement traffic shaping and how this procedure affects resource utilization.

Consider a real-time stream  $v$ . We propose the following traffic-shaping procedure. The source selects some duration  $T$  that is a fraction of the acceptable delay of the traffic in the network. The source is equipped with a buffer whose input is  $v$ . The source reads out its buffer at a rate  $r(t)$  that is equal to the average rate of  $v$  during the interval  $[t - T, t]$ . One can show that by implementing this procedure, the source delays the stream by at most  $T$ . Moreover, the rate  $r(t)$  tends to be more regular (less bursty) than  $v$ .

The traffic shaping delays the cells at the source instead of having them buffered by the network. The rationale is that the network should be used to transport cells and not to store them. An advantage of traffic shaping is that it prevents one stream from perturbing other streams excessively.

#### 8.4.4 Deterministic or Statistical?

One important debate among network researchers is whether networks should use deterministic or statistical procedures. We examine various aspects of that debate in this section. We start with a numerical evaluation based on what we learned in the previous sections.

##### *A Comparison*

We try to compare the number of connections that can be accepted by using the statistical and the deterministic approaches. The comparison is made on a hypothetical model of a video source.

As a numerical example, let us imagine video connections with a mean rate of 1.5 Mbps that can produce bits at 9 Mbps for random periods with durations of up to 10 s corresponding to active movie scenes. We model such a source as being produced by a line rate of 9 Mbps with a maximum burst of 10 s. Thus, for such a source,  $R = 9 \text{ Mbps}$ ,  $R/SCR = 6$ , and the maximum number of back-to-back cells  $M$  is such that  $M/R = 10 \text{ s}$ , that is,  $M \approx 2.1 \times 10^5$  cells. We assume an acceptable delay of 100 ms, that is,  $9 \times 10^5 / 424 = 2,100$  cell transmission times at the line rate  $R$ . From (8.7) we find that the effective bandwidth of this source is very close to 1 and that the source must be treated by the switch as a constant rate source with rate 9 Mbps. Thus, if the output line rate of the switch is 155 Mbps, the number of video connections that can be accepted with a delay of 100 ms is approximately  $155/9 = 17$ .

This example points to the conservatism of this admission strategy. If the source could be viewed as having a rate equal to their mean rate 1.5 Mbps, more than 100 connections could be accepted. If we want to make sure that no cell is delayed by more than 100 ms, then we cannot accept more sources than if they had their peak rate of 9 Mbps. Indeed, it is possible, although very unlikely, that the sources that we accept keep their peak rate for 10 s. If we accept  $N$  sources, then the peak rate is  $9 \times 10^6 \times N$  for 10 s and during that time, the buffer accumulates  $10(9N - 155) \times 10^6$ , which exceeds  $155 \times 10^5$ , that is, 100 ms of buffering, as soon as  $N$  exceeds 17.

We now examine the multiplexing approach. To evaluate the number of sources that the network can accept, we assume that the sources have two rates: 9 Mbps with probability 0.12 and 0.5 Mbps with probability 0.88. This model is approximate but may not be unreasonable. In an actual implementation, the network would measure its spare capacity. With this model we can use the Bahadur-Rao approximation or the binomial distribution as follows. Let us define the random variables  $Y_n$  for  $n \geq 1$  as being iid with  $P(Y_n = 1) = 0.12 = 1 - P(Y_n = 0)$ . We choose some number  $c$  and we find the largest value  $N(c)$  of  $N$  such that

$$P\left(Y_1 + \dots + Y_N > \frac{Nc}{8.5}\right) = 10^{-8}.$$

In addition, we want  $N(c + 0.5) \leq 155$ . The justification is that the rate of source  $n$  is modeled by  $8.5Y_n + 0.5$ . We then maximize  $N(c)$  over  $c$ . The solution to the exercise is  $N = 34$ .

Thus, it is plausible that a statistical method that accepts a small level of risk (a few bad seconds every few days or weeks) can double the number of connections carried by the switch.

### ***Pros and Cons of Deterministic Approaches***

The deterministic approaches based on leaky buckets have the following advantages:

- ◆ the source can easily ensure that its traffic meets the specifications,
- ◆ the network can easily verify that the traffic meets the specifications,
- ◆ the network can guarantee hard bounds on delays and avoid all losses because of buffer overflows, and
- ◆ because the quality of service is specified in terms of hard bounds, the users can verify that the network provides the requested quality of service.

The main disadvantage of deterministic approaches is that they are based on worst cases: worst  $T$  seconds of a stream. The method assumes that all the connections that go through any one node will exhibit their worst-case behavior during the *same*  $T$ -second period.

### ***Difficulties with Statistical Approaches***

The enforcement and policing of statistical traffic descriptions poses unresolved issues. The network uses some quantities for admission and routing of real-time traffic and others for interactive traffic. For real-time traffic, the

network needs to know the parameters of the Bahadur-Rao bound (8.13). The network also needs to know the effective bandwidth and the decoupling bandwidth of the traffic. For interactive traffic, the network needs to know the effective and decoupling bandwidths of the traffic. These quantities are not readily available.

Although we can develop procedures to measure the necessary statistics, it is not clear that we should ask the users to perform these measurements, although protocols for doing so are conceivable. It is likely that repeated video-conferences with the same equipment will have similar statistics. Also, movies that are distributed for video-on-demand applications could remember their statistics. However, all this bookkeeping appears rather cumbersome, and it is tempting to look for solutions that make it unnecessary.

### *A Proposal*

We propose an approach that has the advantages of both the deterministic and statistical approaches. This approach makes simple enforcement and policing possible, and it also permits efficient resource utilization.

We propose that users specify deterministic bounds, such as *PCR*, *SCR*, *BT*, and *CDVT*, for their traffic and statistical descriptions of the quality of service.

The network admits and routes a new connection by assuming worst-case behavior. The network then measures the statistics of the traffic: effective and decoupling bandwidths and the Bahadur-Rao parameters if it is a real-time connection. The network then uses these statistics to keep track of the resources that are now occupied by the ongoing connections.

To prevent the users from always behaving in the worst possible way, we propose that the billing be based on the actual resources they use. In this way, the resource utilization is charged fairly to the different users.

There is one question that this proposal does not answer: how do users verify that they are getting the statistical quality of service that they requested?

## 8.5

## SUMMARY

Virtual circuit networks like ATM seek to combine the gains from statistical multiplexing that packet switching creates with the guaranteed performance that circuit switching offers. When they succeed in this goal, virtual circuit networks will be able to reap the benefits of economies of scale, service integration, and network externalities. In order to succeed, however, network

engineers must solve a number of control problems concerning admission, routing, flow and congestion, and resource allocation.

Problems of routing and admission are similar to those that occur in circuit-switched networks; flow- and congestion-control problems arise in packet-switched networks. And we reviewed the approaches developed in the context of circuit and packet switching.

Problems of resource allocation are peculiar to virtual circuit networks. (In circuit-switched networks each connection gets a fixed resource, whereas in datagram networks no resources are allocated to a connection.) The major difficulty in designing resource-allocation mechanisms is that they depend on (1) characteristics of the user traffic, (2) the available resources, and (3) the quality of service guaranteed to the user. Of these three items, the first is the most difficult to characterize, and we presented two approaches.

The deterministic approach is being pursued by the ATM Forum. It is easy to implement, but its worst-case assumptions will give very conservative answers, leading to significant underutilization of the network. The statistical approach at the present time is still being developed in research laboratories. It is far from standardization, but progress is rapid. We have described the most important accomplishments of both approaches.

The discussion of the statistical approach does make use of the concepts and framework of probability theory. Without an introduction to probability theory, the reader will find it difficult to follow the discussion. However, we have tried to minimize technical concepts so that the material is accessible. The full range of technical detail is concentrated in Chapter 9. That chapter is meant only for the reader with a strong background in probability.

---

## 8.6

## NOTES

For a more detailed discussion of routing in circuit-switched networks, see [K94]. The gradient projection algorithm of section 8.3.3 is studied in [BG92].

For an analysis of the window congestion-control scheme described in section 8.3.4, see [J88, FJ91, J96, MW98]. The receiver-driven rate-based congestion control for the Internet is proposed in [GCMW99]. For a study of congestion control for multicast, see [GS99]. RED was presented in [FJ93].

For details of the ATM Forum's recommendations summarized in section 8.4.2, see [A93, A96d].

A more detailed analysis of the pricing model of section 8.4.2 can be found in [CWW96]. The formulas (8.11) and (8.12) are derived in [AMS82].

The Bahadur-Rao theorem appears in [BR60]. Our discussion is based on [H95]. There is by now a significant literature using large deviations theory to study buffer overflow probabilities and to calculate effective bandwidth; see [W86, Hu88, B90, K91, GH91, CW96, DZ93, DV93, KWC93]. The small buffer analysis is developed in [SW95]. Theorem 8.4.1 is obtained independently in [CW96] and [BD94]. Admission control procedures based on effective bandwidth and decoupling bandwidth are presented in [HW94]. An introduction with an extensive bibliography to this material is given in [W95].

## 8.7

## PROBLEMS

1. What are reasonable delays for interactive database queries, for remote control of a video server, for videoconferences, for telephone conversations, and for transferring X rays?
2. Consider the transmission of a video program with a rate of 1.5 Mbps. Assume that the cell loss rate is equal to  $10^{-8}$ . What is the average time between losses? What is the probability that the transmission of a one-hour video will not have any error?
3. What is the transmission rate required to transmit in 1 s a 4-inch by 6-inch photograph with a resolution of 1,200 dots per inch and 8 bits per pixel?
4. Assume a typical Web surfer makes 1-MB requests at random times at an average rate of 20 requests per day. Assume also that there are 10 million Web surfers and that they access 10,000 Web servers. To simplify, we make the gross assumptions that these servers are equally likely to be consulted. What is the average rate with which one server serves 1-MB requests? What is the minimum connection bandwidth and throughput of a server that is 100 times more popular than average? How would caching the information in distributed servers help run this application?
5. Consider the exchange of important data over a network with a typical end-to-end delay of 5 s and an average transmission time of 30 Kbps. Discuss the effects of a cell error rate of  $10^{-4}$  on such an application. What is the probability that a 1-MB file will be corrupted by transmission errors? Assume that errors are checked for each block of 64 KB in this file. How many blocks are likely to be corrupted? What is the average time until the file is successfully received? How would this time change if the error checking were performed only for the complete file?

*Hint:* To solve this problem, you need to know that if a coin has probability  $p$  of landing on heads in any one flip, then it takes on the average  $1/p$  coin flips until the coin first lands on heads.

6. Consider a large office building with 1,000 telephone sets. Assume that an employee uses a telephone about 30 minutes during a typical 8-hour business day. What is the average number of telephone calls ongoing at a typical time during the business day? Assume that 15% of these calls are outside calls. Give a rough estimate of the number of outside lines that are required.
7. Consider the transmission of messages over a transmission line equipped with a buffer. The messages have lengths that are exponentially distributed with mean  $L$  bits. The transmitter has rate  $C$  bps. The messages arrive as a Poisson process with rate  $\lambda$  messages per second.
  - (a) Using formula (8.1), find the average delay per message. For  $\lambda = 10/\text{s}$  and  $L = 8 \times 10^6$ , find the minimum value of  $C$  so that the average delay does not exceed 0.1 s.
  - (b) How does the average delay per packet change if  $C$  and  $L$  are multiplied by the same constant?
  - (c) Let us pretend that our queue models a Web server that answers requests. We decide to divide up the files in the server into subfiles that are  $K$  times smaller. As a result, the rate of requests for the subfiles becomes  $K\lambda$ . What is the new average delay per subfile?
8. Consider the network of Figure 8.11. Assume that we can choose the parameters  $(p, \mu_1, \mu_2)$  subject to  $\mu_1 + 4\mu_2 \leq 2\lambda$ . Find the values of the parameters that minimize the average delay per packet.
9. As in the previous problem, consider the network of Figure 8.11. Describe a possible implementation of the distributed-gradient routing algorithm for this network. Explain the estimates that the nodes must perform and the information they must exchange.
10. Window congestion control is claimed to be inefficient for connections with a large bandwidth  $\times$  delay product. We discuss a simple model that traces the inefficiency to the long feedback delay. Imagine an M/M/1 queue fed by two streams with respective rates  $\lambda$  and  $\alpha$ . The service rate is  $\mu$ . The average delay through the queue is  $1/r$ , where  $r = \mu - \lambda - \alpha$ . We assume that  $\lambda$  is controlled after a feedback of  $T$  time units. We also assume that the delay through the queue stabilizes after one unit of time. The feedback algorithm is as follows. At time  $t$ , the measured delay is  $r(t) = \mu - \lambda(t-1) - \alpha(t-1)$ . The target delay is  $1/r$ , which corresponds

to a target value of  $\mu - \lambda - \alpha$ . If  $r(t) \neq r$ , then the feedback informs the source to decrease  $\lambda$  by  $r - r(t)$ . This message reaches the source at time  $t + T$ . If the source is aware of the delay, it replaces  $\lambda$  by  $\lambda(t) - r + r(t)$ . Thus,

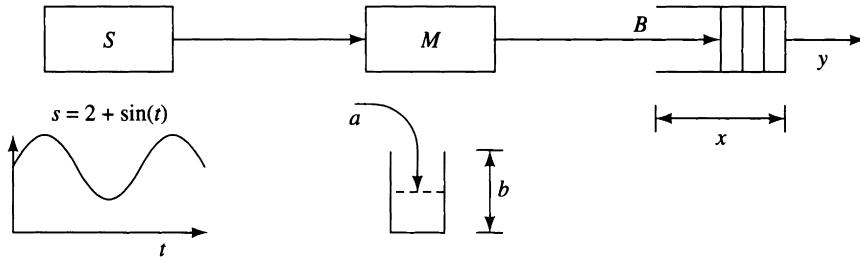
$$\lambda(t + T) = \max\{0, \lambda(t) - r + \mu - \lambda(t - 1) - \alpha(t - 1)\}.$$

Try this algorithm with  $r = 1$ ,  $T = 10$ ,  $\mu = 3$ ,  $\lambda(t) = \alpha(t) = 1$  for  $t \leq 20$ ,  $\alpha(21) = 2$ , and  $\alpha(t) = 1$  for  $t \geq 22$ . Propose methods to stabilize the algorithm.

11. Consider the following analogy to a call admission problem. An elevator can carry 2,000 kg. We want to decide an admission mechanism for people to get into the elevator. The rule of the game is that once somebody is in we cannot ask him or her to leave. Assume that everybody weighs less than 125 kg. The first procedure is to admit 16 people, assuming the worst case. The second procedure is based on statistics that give us the distribution of the weight of an arbitrary person. For simplicity, assume that the weights are Gaussian with mean 70 and standard deviation 15.
  - (a) Find the number of persons that can be admitted so that the probability that the elevator is overloaded does not exceed  $10^{-5}$ .
  - (b) The third method consists in measuring the total load of the elevator as people get in and closing the doors when that load exceeds 1,875 kg. Estimate the expected number of people that can get into the elevator using the same assumptions as in part (a).
12. Write a program to simulate the GCRA( $T, \tau$ ) algorithm and verify the graphs in Figures 8.20 and 8.21.
13. Please refer to the figure on the next page. Let's assume that we have a time-varying traffic source  $S$ . This source generates a "flow" (instead of cells or packets; we do this for simplicity) with an intensity according to  $s = 2 + \sin(t)$  bps. The flow travels through a measurement device  $M$ . This device calculates the leaky-bucket state (i.e., the water level). The leaky bucket has parameters  $a$  (credit arrival rate in bps) and  $b$  (bucket size in bits). The measurement device just measures the traffic as it passes through (i.e., it does not buffer, delay, or discard "nonconforming" traffic). The traffic then enters a buffer  $B$ . This buffer has capacity  $x$  (bits) and is emptied at the constant rate  $y$  (bps).

We will call the leaky-bucket state  $w(t)$  and the buffer occupancy  $o(t)$ . The initial conditions for these states are  $w(0) = b$  and  $o(0) = 0$  (i.e., the bucket is "full" and the buffer is empty.)

- (a) Let  $a = y = 2$  and  $b = x = 2$ . Plot  $s(t)$ ,  $w(t)$ , and  $o(t)$ . How is the leaky-bucket state related to the buffer occupancy?
- (b) Let  $a = y = 2$  and  $b = x = 1/2$ . Plot  $s(t)$ ,  $w(t)$ , and  $o(t)$ . (Note: here we have the same input process and output link rate as in part (a) above yet we have less buffer space, therefore we should expect buffer overflow.) What value of  $y$  must we choose to avoid buffer overflow?
- (c) Express the parameter  $y$  as a function of  $x$  that is required to avoid buffer overflow.
- (d) Suppose we had  $n_1$  traffic sources of type 1 and  $n_2$  traffic sources of type 2. Each traffic source is random. The type 1 traffic sources each "conform" to a leaky-bucket traffic model with parameters  $a_1$  and  $b_1$ . Similarly, the type 2 traffic sources have parameters  $a_2$  and  $b_2$ . If all these traffic sources are to share a single buffer and link, what size buffer and link data rate are required to avoid buffer overflow?



14. Consider a square-wave traffic source  $s(t) = 2$  if  $0 \leq t < 1/2$  or  $1 \leq t < 3/2$  or  $2 \leq t < 5/2$ , and so on, and  $s(t) = 0$  otherwise. The traffic from this source passes by a leaky-bucket measurement device (with parameters  $b$  bits and  $a$  bps) and then into a buffer/link (with parameters  $x$  bits and  $y$  bps).
- What range of link speeds  $y$  (or  $a$ ) result in a large but finite buffer not overflowing?
  - Over that range of link speeds, write an expression for  $x$  (or  $b$ ) for the minimum buffer required to avoid buffer overflow.
  - Let  $y = 1.5$  and  $x = 0.25$ , sketch  $s(t)$ ,  $w(t)$ , and  $o(t)$  (i.e., the source rate, the leaky-bucket state, and the buffer state).
  - Sketch the output of the buffer for the above parameters.
  - Calculate the *average delay* of  $s(t)$  as it passes through the buffer. You should leave  $x$  and/or  $y$  as parameters.

# Control of Networks: Mathematical Background



In this chapter you will learn the mathematical analyses that underlie the control techniques used and the resulting network performance measures described in Chapter 8. Some sections of this chapter demand from the reader a sophisticated background in stochastic processes. A basic knowledge of multivariate random variables and Markov chains is essential for these sections. However, you can skip these sections and still find accessible the discussion on deterministic models. If you are able to follow the more mathematical material, you will be able to participate in the mathematical discussions on networking published in the research journals. But even if you are unable to follow the argument, Chapter 8 makes the conclusions of those discussions accessible.

We start by reviewing some key results on Markov chains in section 9.1. We apply these results to the study of circuit-switched networks in section 9.2 and of datagram networks in section 9.3. Section 9.4 explains the analysis of virtual circuit networks.

---

## 9.1

## MARKOV CHAINS

In this section, we review Markov chains and discuss some key results.

### 9.1.1

### Overview

A Markov chain is a model of the random motion of an object in a discrete set of possible locations. Two versions of this model are of interest to us: discrete

time and continuous time. In discrete time, the position of the object—called the *state* of the Markov chain—is recorded every unit of time, that is, at times 0, 1, 2, and so on. In continuous time, the state is observed at all times  $t \geq 0$ . One can think of the continuous-time model as being a discrete time model where the time unit is infinitesimally small. The state of the Markov chain changes randomly. In discrete time, there is a die at every location. Every time unit, the Markov chain tosses the die at its current location to decide where to jump next. In that way, the law of the future motion of the state depends only on the present location and not on previous locations. This key property that the Markov chain has of “forgetting” its past locations greatly simplifies the analysis.

Engineers use Markov chains to model the progression of the calls that a telephone network carries and of packets that a datagram or virtual circuit network transports. The randomness in these models reflects the uncertainty about when users place calls or send packets and about the length of packets and their destination. The randomness also captures the transmission errors and failures of devices.

The theory of Markov chains tells us how to calculate the fraction of time that the state of the Markov chain spends in the different locations. Network engineers use that theory to estimate the delays and losses of packets in networks or the fraction of time that telephone calls are blocked because all the circuits are busy. The engineers then use these estimates to design and control networks, as we explained in Chapter 8.

In this section, we review the main results of the theory of Markov chains, and we illustrate these results with examples.

### 9.1.2

### Discrete Time

One is given a set  $X$ , called the *state space*. We call the elements of  $X$  *states*. The set  $X$  is countable. That is,  $X$  is either a finite set  $X = \{i_1, \dots, i_N\}$  (for some finite number  $N$ ), or  $X$  is infinite but we can enumerate its elements exhaustively as  $X = \{i_1, i_2, i_3, \dots\}$ . For instance, the set of nonnegative integers  $Z_+ = \{0, 1, 2, 3, \dots\}$  is countable, but the set of real numbers between 0 and 1, that is, the interval  $[0, 1]$ , is not countable. For our purpose, the relevance of a set being countable is that if the elements in a collection  $A$  of positive real numbers add up to 1, then the collection must be countable.

We denote typical elements of  $X$  as  $i, j, k$ . For  $i \in X$  we are given a list of nonnegative numbers  $\{P(i, j), j \in X\}$  that add up to 1. That is,

$$0 \leq P(i, j) \leq 1 \text{ for all } i, j \in X \text{ and } \sum_{j \in X} P(i, j) = 1 \text{ for all } i \in X.$$

We think of  $P = \{P(i, j), i, j \in X\}$  as a matrix. This matrix is  $N$  by  $N$  if  $X$  is finite and has  $N$  elements. If  $X$  is infinite, then the matrix  $P$  is infinite. The matrix  $P$  is called a *transition probability matrix*.

We are also given a probability distribution  $\pi_0$  on  $X$ , that is, a collection of nonnegative numbers  $\{\pi_0(i), i \in X\}$  that add up to 1. That is,

$$0 \leq \pi_0(i) \leq 1 \text{ for all } i \in X \text{ and } \sum_{i \in X} \pi_0(i) = 1.$$

We now define a random sequence  $x = \{x_0, x_1, x_2, \dots\}$  that takes values in  $X$  as follows:

$$P(x_0 = i_0, x_1 = i_1, \dots, x_n = i_n) = \pi_0(i_0)P(i_0, i_1)P(i_1, i_2) \times \dots \times P(i_{n-1}, i_n) \quad (9.1)$$

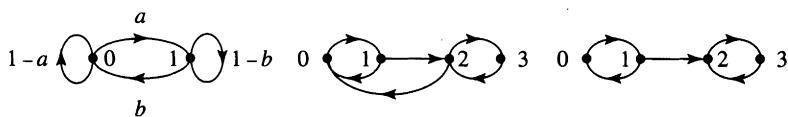
for all  $n \geq 0$  and all  $i_0, i_1, \dots, i_n \in X$ .

The random sequence  $x$  is called a Markov chain with transition probability matrix  $P$  and initial distribution  $\pi_0$ .

The definition of  $x$  specifies that  $x_0$  is selected in  $X$  according to the probability distribution  $\pi_0$  and that if  $x_n = i$ , then  $x_{n+1} = j$  with probability  $P(i, j)$ , independently of the values  $x_m$  for  $m < n$ . The interpretation is that  $x_n$  is the position at time  $n$  of some object that moves randomly in the set  $X$ . If  $x_n = i$ , then the object picks up a die located at state  $i$  and rolls the die. With probability  $P(i, j)$ , the outcome of the roll is  $j$  and the object moves to position  $j$  at time  $n + 1$ .

Figure 9.1 illustrates three Markov chains. The leftmost diagram shows two states, 0 and 1. The arrow from 0 to 1 is marked with the letter  $a$ , which represents some number in  $(0, 1)$ . The meaning of that arrow is that  $P(0, 1) = a$ . Similarly, the other arrows mean that  $P(0, 0) = 1 - a$ ,  $P(1, 0) = b = 1 - P(1, 1) \in (0, 1)$ . Such a diagram is called a *state transition diagram*, and it specifies the transition probability matrix of some Markov chain on  $X = \{0, 1\}$ .

The two other diagrams in Figure 9.1 are other state transition diagrams. By convention, if there is an arrow from  $i$  to  $j$ , then  $P(i, j) > 0$ . The state transition



9.1

State transition diagrams of three discrete-time Markov chains.

diagram is a directed graph. A *path* in such a graph is a succession of arrows such that the end of one arrow is the start of the next arrow. A path corresponds to a possible trajectory of the Markov chain in the state space.

We can calculate the probability distribution of  $x_n$  as follows. For  $n = 1$  we have

$$\pi_1(i) := P(x_1 = i) = \sum_{j \in X} P(x_0 = j, x_1 = i) = \sum_{j \in X} \pi_0(j)P(j, i). \quad (9.2)$$

Define  $\pi_n(i) := P(x_n = i)$  for  $i \in X$  and let  $\pi_n$  be the row vector with elements  $\{\pi_n(i), i \in X\}$ . We can rewrite (9.2) as

$$\pi_1 = \pi_0 P$$

where the right-hand side denotes the product of the row vector  $\pi_0$  by the matrix  $P$ . When  $X$  is infinite, the matrix multiplication involves infinite sums.

By repeating the argument that led us to (9.2) you can verify that

$$\pi_{n+1} = \pi_n P, \quad n \geq 0. \quad (9.3)$$

From this identity we conclude that

$$\pi_n = \pi_0 P^n \quad (9.4)$$

where  $P^n$  is the  $n$ th power of the matrix  $P$  defined as the product of  $P$  by itself  $n$  times.

For instance, with the transition matrix of the leftmost diagram of Figure 9.1,

$$P = \begin{bmatrix} 1 - a & a \\ b & 1 - b \end{bmatrix}, \quad (9.5)$$

we find

$$P^n = \begin{bmatrix} 1 - a_n & a_n \\ b_n & 1 - b_n \end{bmatrix},$$

where

$$a_n = \frac{a + b(1 - a - b)^n}{a + b} \text{ and } b_n = \frac{b + a(1 - a - b)^n}{a + b}.$$

We say that a probability distribution  $\pi$  is *invariant* for the transition probability matrix  $P$  if

$$\pi = \pi P. \quad (9.6)$$

The equations (9.6) are the *balance equations* for the transition probability matrix  $P$ . Note that if  $\pi$  is invariant for  $P$  and if  $\pi_0 = \pi$ , then (9.4) implies that  $\pi_n = \pi$  for all  $n \geq 0$ .

For the transition probability matrix  $P$  given in (9.5) we find that the only solution  $\pi = [\pi(0), \pi(1)]$  of the balance equations (9.6) such that  $\pi(0) + \pi(1) = 1$  is

$$\pi = \left[ \frac{b}{a+b}, \frac{a}{a+b} \right]. \quad (9.7)$$

Before describing the main results about discrete-time Markov chains, we must introduce a few definitions.

**Definition 9.1.1** Let  $P$  be a transition probability matrix on the state space  $X$ . The transition matrix  $P$  is *irreducible* if it is possible for a Markov chain with transition matrix  $P$  to move from any state  $i$  to any other state  $j$  in finite time. In other words,  $P$  is irreducible if there is a path from every  $i$  to every other  $j$  in the state transition diagram that corresponds to  $P$ .

A Markov chain with transition probability matrix  $P$  is said to be irreducible if  $P$  is irreducible.

For instance, looking at the state transition diagrams of Figure 9.1, we find that the two leftmost Markov chains are irreducible, whereas the rightmost is not.

The following theorem tells us about the possible invariant distributions of an irreducible Markov chain.

**Theorem 9.1.1** An irreducible Markov chain has at most one invariant distribution. It certainly has one if it is finite. The Markov chain is said to be positive recurrent if it has one invariant distribution.

We noted earlier that the leftmost Markov chain of Figure 9.1 has a unique invariant distribution that is given by (9.7). The theorem tells us that the Markov chain in the center of Figure 9.1 also has a unique invariant distribution.

The invariant distribution, when it exists, measures the fraction of time that the Markov chain spends in the various states. This relationship is expressed in the following theorem.

**Theorem 9.1.2** Let  $x = \{x_n, n \geq 0\}$  be an irreducible Markov chain on  $X$ . Then

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} 1\{x_n = i\} = \pi(i), \quad i \in X, \quad (9.8)$$

where  $\pi$  is the unique invariant distribution of the Markov chain if it exists and  $\pi(i) := 0$  for  $i \in X$  if the Markov chain has no invariant distribution.

In the identity (9.8), the notation  $1\{x_n = i\}$  has the value 1 if  $x_n = i$  and the value 0 otherwise. Thus, the quantity

$$\frac{1}{N} \sum_{n=0}^{N-1} 1\{x_n = i\}$$

is the fraction of time that the Markov chain spends in state  $i$  during the first  $N$  units of time. This fraction of time is random because it depends on the particular realization of the sequence  $x$  that happens to occur. The theorem says that, in the long term, this fraction of time approaches a nonrandom quantity  $\pi(i)$ .

Thus, if the Markov chain has an invariant distribution  $\pi$ , then  $\pi(i)$  is the long-term fraction of time that the Markov chain spends in state  $i$ , for  $i \in X$ . If the Markov chain does not have an invariant distribution, then the fraction of time that the Markov chain spends in any one state is negligible. What happens in that case is either that the Markov chain is wandering off to infinity, if one enumerates the states in any arbitrary way, or that it visits all the states infinitely often but makes such large excursions in the state space that it spends a negligible fraction of time in any finite set of states.

The following theorem gives another important interpretation of the invariant distribution. However, we need one more definition to state that result.

Let  $P$  be the transition probability matrix of an irreducible Markov chain. Define

$$d(i) = \gcd\{n \geq 1 | P^n(i, i) > 0\}, i \in X. \quad (9.9)$$

In this definition,  $\gcd A$  denotes the greatest common divisor of the elements of the set  $A$ . For instance,  $\gcd\{6, 9, 12, 15, \dots\} = 3$  and  $\gcd\{1, 4, 6, 8, 10, 12, \dots\} = 1$ . The set in (9.9) is the collection of numbers of steps  $n$  such that the Markov chain can go from the state  $i$  back to itself in  $n$  steps.

For instance, consider the leftmost Markov chain of Figure 9.1 and assume that  $0 < a < 1$  and  $0 < b < 1$ . One finds that  $\{n \geq 1 | P^n(0, 0) > 0\} = \{1, 2, 3, 4, \dots\}$  so that  $d(0) = \gcd\{1, 2, 3, 4, \dots\} = 1$ . Similarly, one can verify that  $d(1) = 1$ . Now assume that  $a = b = 1$ . Then one finds  $d(0) = \gcd\{2, 4, 6, 8, \dots\} = 2$  and  $d(1) = \gcd\{2, 4, 6, 8, \dots\} = 2$ .

It can be proved that, for any irreducible Markov chain,  $d(i) = d$  for all  $i \in X$ . This leads us to the following definition.

**Definition 9.1.2** Let  $P$  be an irreducible probability transition matrix on  $X$ . Define

$$d = \gcd\{n \geq 1 | P^n(i, i) > 0\}, i \in X.$$

If  $d > 1$ , then  $P$  is said to be *periodic with period d*. If  $d = 1$ , then  $P$  is said to be *aperiodic*.

A Markov chain with transition probability matrix  $P$  is also said to be aperiodic or periodic with period  $d$ .

Thus, the leftmost Markov chain in Figure 9.1 is aperiodic if  $0 < \alpha < 1$  and  $0 < b < 1$ ; it is periodic with period 2 if  $\alpha = 1 = b$ . Note that in the latter case, the values  $x_n$  alternate between 0 and 1 for  $n \geq 0$ . Thus, in that case, if  $\pi_0(0) = \alpha$  with  $0 < \alpha < 1$ , then  $\pi_{2n}(0) = \alpha$  for  $n \geq 0$  and  $\pi_{2n+1}(0) = 1 - \alpha$  for  $n \geq 0$ . This example shows that the distribution at time  $n$ ,  $\pi_n$  alternates between two values and therefore does not converge. The periodicity 2 of the Markov chain is reflected in the periodicity of its distribution as a function of time.

The Markov chain in the center of Figure 9.1 is aperiodic. Indeed,  $\{n \geq 1 | P^n(0, 0) > 0\} = \{2, 3, 4, 5, \dots\}$  so that  $d = 1$ .

One may hope that if the Markov chain is aperiodic, the distributions  $\pi_n$  may converge. This is indeed the case, as the following theorem makes precise.

**Theorem 9.1.3** Let  $x$  be an irreducible and aperiodic Markov chain with invariant distribution  $\pi$ . Then, for any initial distribution  $\pi_0$ ,

$$\pi_n(i) \rightarrow \pi(i) \text{ as } n \rightarrow \infty, \text{ for all } i \in X.$$

The above theorem tells us that if we start a Markov chain  $x$  that is irreducible and aperiodic and has an invariant distribution  $\pi$  and if we wait long enough, then the probability of finding the Markov chain in state  $i$  is close to  $\pi(i)$ . The interpretation is that the Markov chain *approaches steady state*.

It is often comforting to be able to show that a Markov chain is positive recurrent even if one is not able to calculate its invariant distribution. For instance, if the Markov chain is the model of a buffer occupancy, then showing that it is positive recurrent tells us that the buffer empties relatively frequently and gives us a sense of the system's stability.

The following result is one of a number of useful sufficient conditions for positive recurrence. This result has the advantage of being intuitive and of being applicable to many queuing systems.

**Theorem 9.1.4** Let  $x$  be an irreducible Markov chain on  $X$  and  $V : X \rightarrow [0, \infty)$  some function. Define the *drift*  $\Delta(x)$  of  $f(\cdot)$  at  $x$  by

$$\Delta(x) := E[V(x_{n+1}) - V(x_n) | x_n = x], \text{ for } x \in X.$$

Assume that there is some *finite* subset  $S$  of  $X$  and some constants  $D > 0$  and  $A < \infty$  such that

$$\Delta(x) \leq -D < 0, \text{ for } x \notin S, \quad (9.10)$$

and

$$\Delta(x) \leq A < \infty, \text{ for all } x. \quad (9.11)$$

Then the Markov chain  $x$  is positive recurrent.

We leave the proof of this result as an exercise. Roughly speaking, if the fraction of time that  $x$  spends in  $S$  were negligible, then the expected value of  $V(x_n)$  would keep decreasing at rate  $-D < 0$  forever. This cannot be since  $V(x) \geq 0$ . Thus,  $x$  must spend a positive fraction of time in the finite set  $S$ . In view of Theorem 9.1.2, it follows that  $x$  must be positive recurrent.

### 9.1.3 Continuous Time

Engineers often use continuous-time models of networks. As we stated in the overview of this section, one can view a continuous-time Markov chain as a discrete-time Markov chain with an infinitesimally small time unit. However, it is easier to work with a more direct definition. Before introducing that definition, we need to recall a few facts about the exponential distribution.

**Definition 9.1.3** The random variable  $\tau$  is *exponentially distributed with rate  $\lambda > 0$*  if

$$P(\tau > t) = e^{-\lambda t}, \text{ for } t \geq 0. \quad (9.12)$$

For convenience, when  $\lambda = 0$ , we define  $\tau$  to be an infinite random variable (i.e.,  $\tau = +\infty$  with probability 1).

The following properties follow from this definition.

**Theorem 9.1.5** Let  $\tau$  be exponentially distributed with rate  $\lambda$ . Then

(a) The mean value  $E(\tau)$  of  $\tau$  is given by

$$E(\tau) = \frac{1}{\lambda};$$

(b) The random variable is *memoryless*. That is,

$$P[\tau > t + s | \tau > s] = P(\tau > t), \text{ for all } s, t \geq 0.$$

The memoryless property can be interpreted as follows. Assume that a light bulb has an exponentially distributed lifetime. Then knowing how old the bulb is does not help predict how long it will still live. In other words, an old bulb is exactly as good as a new one: the bulb does not age. (That is, until it suddenly dies.)

Using exponential distribution we can construct a continuous-time Markov chain. We first define a rate matrix  $Q$ .

**Definition 9.1.4** Let  $X$  be a countable set. A *rate matrix*  $Q$  on  $X$  is a collection  $Q = \{q(i, j), i, j \in X\}$  of real numbers such that

$$0 \leq q(i, j) < \infty, \text{ for all } i \neq j \in X, \text{ and}$$

$$-q(i, i) = q(i) := \sum_{j \neq i} q(i, j) < \infty, \text{ for all } i \in X.$$

We are now ready to define a continuous-time Markov chain.

**Definition 9.1.5** Let  $X$  be a countable set and  $Q$  a rate matrix on  $X$ . Let also  $\pi$  be a probability distribution on  $X$ . We define a continuous-time Markov chain  $x := \{x_t, t \geq 0\}$  on  $X$  with rate matrix  $Q$  and initial distribution  $\pi$  as follows.

First one chooses  $x_0$  with distribution  $\pi$  in  $X$ . That is,  $P(x_0 = i) = \pi(i)$  for  $i \in X$ .

Second, if  $x_0 = i$ , one selects a random time  $\tau$  that is exponentially distributed with rate  $q(i)$ . The process  $x$  is defined so that

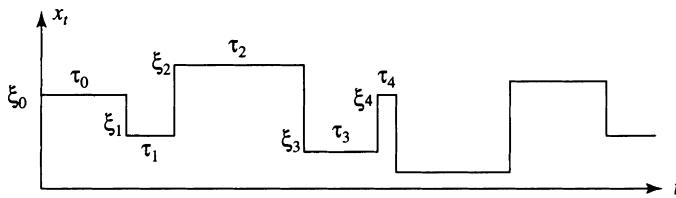
$$x_t = i \text{ for } 0 \leq t < \tau.$$

Third, at time  $t = \tau$ , the process  $x$  makes a jump from its initial value  $i$  to a new value  $j$  that is selected independently of  $\tau$  and so that

$$P[x_\tau = j | x_0 = i, \tau] = \Gamma(i, j) := \frac{q(i, j)}{q(i)}, j \neq i.$$

The construction then resumes from  $x_\tau = j$  at time  $\tau$ , independently of the process before time  $\tau$ .

Figure 9.2 shows a typical realization of the process  $x$ . Thus, the process starts in some state  $\xi_0$  and keeps that value for  $\tau_0$ , then visits a sequence of states



**FIGURE**

Trajectory of the continuous-time Markov chain  $x_t$ . The successive states are  $\xi_0, \xi_1, \xi_2, \dots$ , and the successive holding times are  $\tau_0, \tau_1, \tau_2, \dots$ .

$\xi_1, \xi_2, \xi_3, \dots$ , where it stays for  $\tau_1, \tau_2, \tau_3, \dots$ , respectively. The sequence of successive states is such that

$$P(\xi_0 = i_0, \xi_1 = i_1, \dots, \xi_n = i_n) = \pi(i_0)\Gamma(i_0, i_1)\Gamma(i_1, i_2)\cdots\Gamma(i_{n-1}, i_n).$$

Moreover, given this sequence of states, the successive *holding times*  $\tau_0, \tau_1, \dots, \tau_n$  are independent and exponentially distributed with rates  $q(i_0), q(i_1), \dots, q(i_n)$ , respectively.

We assume in this text that the rate matrix  $Q$  is such that the jump times do not accumulate. That is, we assume that

$$\sum_{n=0}^{\infty} \tau_n = \infty, \text{ with probability 1.}$$

With this assumption, the construction that we described defines a process  $\mathbf{x}$  over  $[0, \infty)$ . The rate matrix  $Q$  is said to be *regular* if it has that property. We always assume that the rate matrices are regular.

The memoryless property of the exponential distribution implies that the Markov chain  $\mathbf{x}$  starts afresh from  $x_t$  at time  $t$ , for any  $t \geq 0$ . That is, we have the following property.

**Theorem 9.1.6** Let  $\mathbf{x}$  be a continuous-time Markov chain with rate matrix  $Q$  on  $\mathbf{X}$ . Then, for any set  $\mathbf{A}$  of trajectories in  $\mathbf{X}$ ,

$$P[(x_s, s \geq t) \in \mathbf{A} | x_t = i; x_u, u < t] = P[(x_s, s \geq 0) \in \mathbf{A} | x_0 = i].$$

The set  $\mathbf{A}$  in the theorem is any set of trajectories of  $\mathbf{x}$  in  $\mathbf{X}$  of the form

$$\mathbf{A} = \{\mathbf{x} | x_{t_k} \in S_k \text{ for } k = 1, \dots, K\},$$

where  $K \leq \infty$ ,  $0 \leq t_1 < t_2 < \dots < t_K$ , and  $S_k$  is a subset of  $\mathbf{X}$  for  $k = 1, \dots, K$ .

This result says that the only information about the trajectory of  $\mathbf{x}$  up to time  $t$  that is useful for predicting the trajectory after time  $t$  is the current value  $x_t$ . The intuitive justification of this theorem is that the past values of  $\mathbf{x}$  are irrelevant because of the way successive values are selected. Moreover, the past holding times are irrelevant because the future ones depend only on the future states. Finally, the elapsed value of the current holding time is irrelevant for predicting when the next jump will occur because that holding time is memoryless.

Next we study the invariant distribution of a continuous-time Markov chain. We first need to define *irreducibility*.

**Definition 9.1.6** A rate matrix  $Q$  on  $X$  is *irreducible* if  $q(i) > 0$  for all  $i \in X$  and if the transition probability matrix  $\Gamma$  defined by

$$\Gamma(i, j) = \begin{cases} \frac{q(i,j)}{q(i)}, & i \neq j \in X \\ 0, & i = j \in X \end{cases}$$

is irreducible.

A continuous-time Markov chain with rate matrix  $Q$  is said to be *irreducible* if its rate matrix  $Q$  is irreducible.

Thus, a continuous-time Markov chain is irreducible if it can go from any state to any other state in finite time. With this definition we can state the main result about continuous-time Markov chains.

**Theorem 9.1.7** Let  $\mathbf{x}$  be an irreducible Markov chain on  $X$  with rate matrix  $Q$  and initial distribution  $\pi$ .

(a) The distribution  $\pi$  is invariant, that is,

$$P(x_t = i) = \pi(i), \text{ for all } i \in X \text{ and } t \geq 0,$$

if and only if  $\pi$  solves the following *balance equations*:

$$\sum_{i \in X} \pi(i) q(i, j) = 0, \text{ for all } j \in X.$$

(b) The Markov chain is stationary if and only if its initial distribution is invariant.

(c) The Markov chain has either no or one invariant distribution. It certainly has one if  $X$  is finite.

(d) If the Markov chain has one invariant distribution  $\pi$ , then

$$\lim_{t \rightarrow \infty} P(x_t = i) = \pi(i), \text{ for all } i \in X, \text{ and}$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T 1\{x_s = i\} ds = \pi(i), \text{ for all } i \in X.$$

(e) If the Markov chain has no invariant distribution, then

$$\lim_{t \rightarrow \infty} P(x_t = i) = 0, \text{ for all } i \in X, \text{ and}$$

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T 1\{x_s = i\} ds = 0, \text{ for all } i \in X.$$

It is often useful to consider a Markov chain reversed in time. The following theorem summarizes the key features of that process.

**Theorem 9.1.8** (a) Let  $x$  be a stationary continuous-time Markov chain on  $X$  with rate matrix  $Q$  and invariant distribution  $\pi$ . Then

$$\tilde{x} := \{x_{T-t}, 0 \leq t \leq T\}$$

is a stationary continuous-time Markov chain on  $X$  with invariant distribution  $\pi$  and with rate matrix  $\tilde{Q}$  given by

$$\tilde{q}(i, j) = \frac{\pi(j)q(j, i)}{\pi(i)}, i, j \in X.$$

The process  $\tilde{x}$  is said to be  $x$  reversed in time.

(b) Let  $x$  be a continuous-time Markov chain on  $X$  with rate matrix  $Q$ . If  $\pi$  is a distribution on  $X$  and  $Q'$  is a rate matrix on  $X$  such that

$$\pi(i)q(i, j) = \pi(j)q'(j, i), \text{ for all } i, j \in X,$$

then  $\pi$  is invariant for  $x$  and  $Q'$  is the rate matrix of  $x$  reversed in time.

The expression for  $\tilde{Q}$  given in part (a) of the theorem can be understood as follows. Assume that  $x$  is stationary with invariant distribution  $\pi$ . Then

$$P(x_0 = i, x_t = j) = \pi(i)q(i, j)t + o(t), \text{ for all } i \neq j \in X.$$

But if  $\tilde{\mathbf{x}}$  is  $\mathbf{x}$  reversed in time, then

$$\begin{aligned} P(\tilde{x}_0 = i, \tilde{x}_t = j) &= \pi(i)\tilde{q}(i, j)t + o(t) \\ &= P(x_t = i, x_0 = j) = P(x_0 = j, x_t = i) \\ &= \pi(j)q(j, i)t + o(t), \end{aligned}$$

which shows that  $\pi(i)\tilde{q}(i, j) = \pi(j)q(j, i)$ .

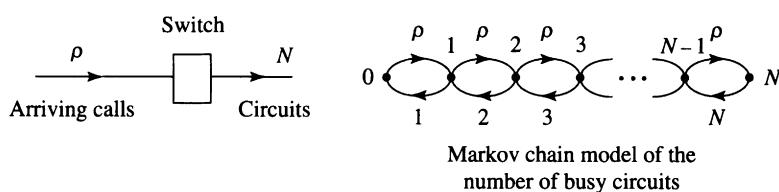
With this rapid review of the main results on Markov chains, we are ready to analyze models of communication networks. We start with circuit-switched networks before moving on to packet-switched networks.

## 9.2 CIRCUIT-SWITCHED NETWORKS

In this section we explain the basic theory of circuit-switched networks. We start by discussing the case of a single switch in section 9.2.1. In section 9.2.2 we examine the case of a network.

### 9.2.1 Single Switch

Consider the situation shown in Figure 9.3. Phone calls arrive at a single switch as a Poisson process with rate  $\rho$ . That is, the times between successive calls are independent and exponentially distributed with rate  $\rho$ . There are  $N$  circuits to carry the calls. If a call arrives when all the lines are busy, then the call is rejected (blocked). The durations of the calls are independent and exponentially distributed with rate 1.



9.3

Model of a circuit switch. Calls arrive at rate  $\rho$  and are carried by  $N$  circuits.

FIGURE

### *Markov Chain Model*

We want to calculate the fraction of calls that are blocked. To perform this calculation, we observe that the number  $x_t$  of calls in progress at time  $t \geq 0$  is a Markov chain with the transition diagram shown in Figure 9.3. In this diagram, an arrow from state  $i$  to state  $j$  is labeled with the value  $q(i, j)$  of the entry of the rate matrix that corresponds to that pair of states.

To explain the transition diagram, assume that there are  $n$  calls in progress at time  $t$ , so that  $x_t = n$ . The next transition of  $\mathbf{x}$  will occur when either a new call is placed or when one of the  $n$  ongoing calls is completed. The time  $\sigma$  until the next call arrives is exponentially distributed with rate  $\rho$ . The time  $\tau$  until the first call in progress terminates is the minimum of  $n$  independent exponentially distributed random variables  $\tau_1, \dots, \tau_n$  that are the residual values of the durations of the calls in progress at time  $t$ . These residual values are exponentially distributed with rate 1 because the original call durations are exponentially distributed and are therefore memoryless, so the residual durations are distributed exactly as the original durations. Now, the minimum  $\tau$  of the random variables  $\tau_1, \dots, \tau_n$  is exponentially distributed with rate  $n$  because

$$\begin{aligned} P(\tau > t) &= P(\tau_1 > t, \tau_2 > t, \dots, \tau_n > t) \\ &= P(\tau_1 > t)P(\tau_2 > t) \cdots P(\tau_n > t) = e^{-t}e^{-t} \cdots e^{-t} = e^{-nt}. \end{aligned}$$

The next transition of  $\mathbf{x}$  after time  $t$  occurs after the minimum of  $\tau$  and  $\sigma$ . This time is exponentially distributed with rate  $\rho + n$  because

$$P(\min\{\tau, \sigma\} > t) = P(\tau > t, \sigma > t) = P(\tau > t)P(\sigma > t) = e^{-nt}e^{-\rho t} = e^{-(n+\rho)t}.$$

Now, when this transition occurs, the probability that it is due to a new call instead of a call termination is given by

$$\begin{aligned} P[\sigma < \tau | \min\{\sigma, \tau\} \in (s, s + \epsilon)] &= \frac{P(\tau > s, \sigma \in (s, s + \epsilon))}{(n + \rho)\epsilon \exp\{-(n + \rho)s\}} \\ &= \frac{[\exp\{-ns\}] \times [\rho\epsilon \exp\{-\rho s\}]}{(n + \rho)\epsilon \exp\{-(n + \rho)s\}} \\ &= \frac{\rho}{n + \rho}. \end{aligned}$$

Thus, the Markov chain  $\mathbf{x}$  is such that

$$q(n) = \rho + n \text{ and } \Gamma(n, n + 1) = \frac{\rho}{n + \rho}.$$

Since  $\Gamma(n, n+1) = q(n, n+1)/q(n)$ , we conclude that  $q(n, n+1) = \rho$  and consequently that  $q(n, n-1) = n$ .

One can understand the diagram of Figure 9.3 more directly than by doing the above calculations. The diagram shows that when  $x_t = n$ , that is, when there are  $n$  calls in progress, a new call arrives with rate  $\rho$ , so that  $x$  jumps from  $n$  to  $n+1$  with rate  $\rho$ . Also, a call terminates, and  $x$  jumps from  $n$  to  $n-1$  with rate  $n$ . This rate  $n$  is the sum of the  $n$  rates of completion of the calls in progress.

### Invariant Distribution

The diagram shows that  $x$  is irreducible. By Theorem 9.1.3, we calculate the invariant distribution of  $x$  by solving the balance equations

$$\sum_{i \in X} \pi(i)q(i, j) = 0, \text{ for all } j \in X.$$

By using the definition  $q(i) = \sum_{j \neq i} q(i, j) = -q(i, i)$ , we can rewrite these equations as

$$\pi(i)q(i) = \sum_{j \neq i} \pi(j)q(j, i), \text{ for all } i \in X. \quad (9.13)$$

The interpretation of the equation (9.13) for a particular value of  $i$  is that the rate of transitions out of state  $i$  should equal the rate of transitions into state  $i$ .

Thus, for the Markov chain of Figure 9.3, the balance equations are as follows:

$$\begin{aligned} \pi(0)\rho &= \pi(1) \\ \pi(1)(\rho + 1) &= \pi(0)\rho + \pi(2)2 \\ &\dots \\ \pi(N)N &= \pi(N-1)\rho. \end{aligned}$$

Remembering that the distribution  $\pi$  must sum to 1, we find that the solution of these balance equations is given by

$$\pi(n) = \frac{\rho^n/n!}{\sum_{m=0}^N \rho^m/m!}, \text{ for } n = 0, 1, \dots, N.$$

In particular, we find that

$$P(x_t = N) = \pi(N) = E(\rho, N) := \frac{\rho^N/N!}{\sum_{m=0}^N \rho^m/m!}.$$

### Erlang Loss Formula

The formula  $E(\rho, N)$  is called the *Erlang loss formula*. It represents the fraction of time that the  $N$  circuits are busy. This fraction of time is also the fraction of the calls that arrive when all the circuits are busy and are therefore blocked. To see why this is the case, we calculate the probability  $\alpha(n)$  that a call that arrives finds  $n$  circuits busy: We find

$$\begin{aligned}\alpha(n) &= P[x_t = n | \text{a call arrives in}(t, t + \epsilon)] \\ &= \frac{P(x_t = n)P[\text{a call arrives in}(t, t + \epsilon) | x_t = n]}{P(\text{a call arrives in}(t, t + \epsilon))} \\ &= \frac{\pi(n)\rho\epsilon}{\rho\epsilon} = \pi(n).\end{aligned}$$

In the last equation, we used the fact that an arrival occurs in the next  $\epsilon$  time units with probability  $\rho\epsilon$ , independently of  $x_t$ , by definition of the arrival process.

This calculation shows that the fraction of calls that find  $n$  circuits busy is equal to the fraction of time that  $n$  circuits are busy. In particular, for  $n = N$ , the fraction of calls that are blocked is the fraction of time that all the circuits are busy. Consequently, the blocking probability is given by the Erlang loss formula.

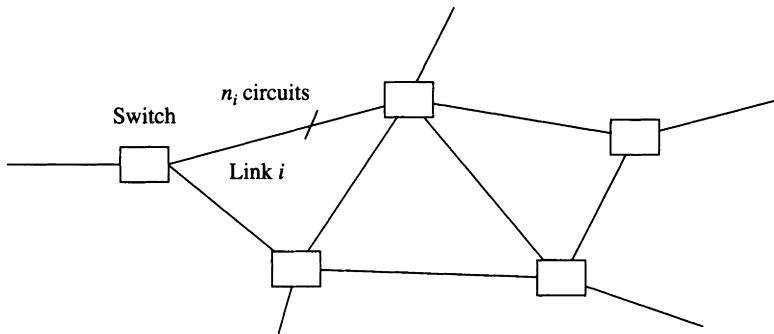
### Insensitivity

We have calculated the blocking probability at a switch by assuming that the call durations are exponentially distributed. It turns out that the blocking probability does not change if the call durations have another distribution with the same mean value.

In other words, the blocking probability is *insensitive* to the actual distribution of the call durations. We explain a proof of that important result in section 9.3.5.

## 9.2.2 Network

Consider the network of Figure 9.4. The network has  $K$  switches that are connected by  $L$  links. Link  $i$  has  $n_i$  circuits. A route is an acyclic set of links. When a call is routed along a route, it uses one circuit in each link along the route.



9.4

FIGURE

Circuit-switched network. The network has  $K$  switches that are interconnected by  $L$  links. Link  $i$  has  $n_i$  circuits.

We assume that calls are placed along route  $r$  as a Poisson process with rate  $\lambda_r$  for  $r = 1, \dots, R$ . The call durations are independent and exponentially distributed with rate 1. We want to analyze the blocking probability of calls.

Denote by  $x_t^r$  the number of calls in progress along route  $r$  for  $r = 1, \dots, R$  and for  $t \geq 0$ . Let  $\mathbf{x}_t = (x_t^r, r = 1, \dots, R)$ . Because of the memoryless property of the exponential distribution, the process  $\mathbf{x} = \{\mathbf{x}_t, t \geq 0\}$  is a Markov chain. The state space  $\mathbf{X}$  of that Markov chain is

$$\mathbf{X} = \{\mathbf{x} \in \mathbb{Z}_+^R \mid \sum_{r=1}^R x^r A(r, i) \leq n_i, \text{ for } i = 1, \dots, L\}.$$

In this expression,  $A(r, i)$  takes the value 1 if route  $r$  goes through link  $i$  and the value 0 otherwise. Thus,  $\sum_{r=1}^R x^r A(r, i)$  is the number of calls that go through link  $i$ . That number must be at most  $n_i$  since link  $i$  has  $n_i$  circuits and can carry at most  $n_i$  calls.

If the links had an infinite number of circuits, then the numbers of calls in progress along different routes would be independent because they would not interfere with one another. In that case, the number of calls in progress along route  $r$  would be modeled by a Markov chain on  $\{0, 1, 2, \dots\}$  with a rate matrix  $Q$  such that  $q(n, n+1) = \lambda_r$  and  $q(n, n-1) = n$ . By solving the balance equations of this Markov chain we would find that its invariant distribution is given by

$$\pi_r(n) = \frac{\lambda_r^n}{n!} e^{-\lambda_r}, \text{ for } n \geq 0.$$

Thus, if the links had infinitely many circuits, the invariant distribution  $\pi$  of  $\mathbf{x}$  would be, by independence,

$$\pi(x^1, \dots, x^R) = \pi_1(x^1) \dots \pi_R(x^R).$$

However, since the links have a finite number of circuits, calls along different routes interfere with one another. Remarkably, the invariant distribution  $\pi_X$  in this case remains proportional to  $\pi$ . That is, the invariant distribution is given by

$$\pi_X(x^1, \dots, x^R) = \frac{\pi(x^1, \dots, x^R)}{\pi(X)}, \text{ for } (x^1, \dots, x^R) \in X,$$

where  $\pi(X) := \sum_{x \in X} \pi(x)$ . Note that the denominator normalizes the distribution so that it adds up to 1 over  $X$ .

To prove this result we denote by  $Q$  the rate matrix of the Markov chain  $\mathbf{x}$  on  $X$ , and we observe that

$$\pi_X(x)q(x, y) = \pi_X(y)q(y, x), \text{ for all } x, y \in X. \quad (9.14)$$

To see this, let  $x \in X$  and  $y = x + e_r$ , where  $e_r$  is the unit vector in direction  $r$ . That is,  $y^i = x^i + 1\{i = r\}$ . Then  $q(x, y) = \lambda_r$  since a transition from  $x$  to  $y$  corresponds to an arrival of a call along route  $r$  and  $q(y, x) = x^r + 1$  since the transition from  $y$  to  $x$  is the completion of a call along route  $r$  when  $x^r + 1$  calls are in progress along that route. Thus, equation (9.14) reads

$$\frac{1}{\pi(X)} \pi_1(x^1) \dots \pi_r(x^r) \dots \pi_R(x^R) \lambda_r = \frac{1}{\pi(X)} \pi_1(x^1) \dots \pi_r(x^r + 1) \dots \pi_R(x^R) (x^r + 1),$$

and we see that this equation is satisfied because of the form of  $\pi_r$ . Other instances of pairs of states  $x$  and  $y$  can be checked along similar lines.

We can use the invariant distribution  $\pi_X$  to calculate the fraction of calls that are blocked. To do this, we consider a call that is placed along route  $r$ . Such a call is blocked if it is placed when the network is in a state  $x \in X$  such that  $x + e_r$  is no longer in  $X$ . Let us denote the set of such states by  $X_r$ . The fraction of time that the network state is in  $X_r$  is  $\pi_X(X_r)$ . Arguing as we did for a single switch, we can show that the fraction of calls placed along route  $r$  that find the network state in  $X_r$  is equal to the fraction of time  $\pi_X(X_r)$  that the network state is in that set. Thus, the blocking probability  $B_r$  for a call placed along route  $r$  is given by

$$B_r = \pi_X(X_r) = \frac{\pi(X_r)}{\pi(X)}. \quad (9.15)$$

### *Complexity Considerations*

In principle, the network engineers can use these formulas to calculate blocking probabilities and rates of revenues of the network, as we discussed in Chapter 8. However, the calculation of the numerator and denominator in (9.15) is very time-consuming because of the large number of elements in the sets  $X$  and  $X_r$ . That number of elements is of the order of

$$\prod_{i=1}^L n_i.$$

To reduce the complexity of the calculations, it is possible to develop recursions in the  $n_i$ . It is also possible to evaluate the numerator and denominator of (9.15) by Monte Carlo simulations. The idea is to generate at random a vector  $x$  according to the distribution  $\pi$ . This generation is made particularly simple by the product form of  $\pi$ . By repeating the experiment we then estimate  $\pi(X)$  as the fraction of the samples  $x$  that happen to fall in  $X$  and similarly for  $\pi(X_r)$ . A simple calculation can be used to estimate the number of random samples that should be generated in order to estimate the blocking probability within a few percent with a high degree of confidence (say 95%). This simulation can also be speeded up by exploiting importance sampling methods.

### *Erlang Fixed Point*

To circumvent the numerical complexity of evaluating (9.15), researchers have developed an approximation called the *Erlang fixed-point approximation*. This approximation is based on assuming that a call is blocked independently by the different links along its route. With this assumption, a call along route  $r$  will not be blocked and will appear on link  $i$  along route  $r$  with probability

$$\Pi_{i \neq j \in r} (1 - B_j),$$

where  $B_j$  is the probability that link  $j$  blocks the call and where the product is over all the links  $j$  other than  $i$  along route  $r$ . Thus, the rate  $\lambda_r$  of calls along route  $r$  contributes a rate

$$\lambda_r \times \Pi_{i \neq j \in r} (1 - B_j)$$

of calls on link  $i$  since this rate is the rate of calls that are not blocked by other links along route  $r$ . Thus, by summing over the different routes  $r$  we find that the rate  $\rho_i$  of calls that are placed on link  $i$  is given by

$$\rho_i = \sum_{r=1}^R \lambda_r \times \Pi_{i \neq j \in r} (1 - B_j). \quad .$$

Now, the blocking probability  $B_j$  at link  $j$  is a function of the rate  $\rho_j$  of calls on that link. In fact,  $B_j = E(\rho_j, n_j)$  where  $E(\rho, n)$  is the Erlang loss formula for  $n$  circuits faced by a Poisson process of calls with rate  $\rho$ . Thus, we find that

$$\rho_i = \sum_{r=1}^R \lambda_r \times \prod_{i \neq j \in r} (1 - E(\rho_j, n_j)), \text{ for } i = 1, \dots, L. \quad (9.16)$$

These equations form a set of fixed-point equations that define the rates  $\rho_i$  in terms of themselves. These equations are the *Erlang fixed-point equations*. It has been shown that these equations provide a good approximation of the blocking probabilities when the network is large.

## 9.3 DATAGRAM NETWORKS

In this section we apply the theory of continuous-time Markov chains to the analysis of datagram networks. In the process, we derive the key results on product-form queuing networks.

### 9.3.1 M/M/1 Queue

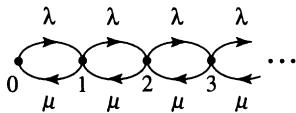
An *M/M/1 queue* is a waiting room equipped with a service facility where customers arrive as a Poisson process with rate  $\lambda$ ; the customers are served by a single server, and their service times are independent and exponentially distributed with rate  $\mu > \lambda$ . The first M in the notation M/M/1 means that the arrival process is memoryless (Poisson); the second M means that the service times are memoryless (exponentially distributed). The 1 indicates that there is a single server.

Because of the memoryless property of the exponential distribution, the number  $x_t$  of customers in the queue at time  $t$ , including the one in service, is a Markov chain with the transition diagram shown in Figure 9.5. The balance equations for the Markov chain are as follows:

$$\begin{aligned}\pi(0)\lambda &= \pi(1)\mu \\ \pi(n)(\lambda + \mu) &= \pi(n-1)\lambda + \pi(n+1)\mu \text{ for } n \geq 1\end{aligned}$$

The solution of these equations is easily verified to be

$$\pi(n) = (1 - \rho)\rho^n, n \geq 0, \text{ with } \rho := \frac{\lambda}{\mu}. \quad (9.17)$$



**9.5**  
**FIGURE**

Markov chain model of an M/M/1 queue with arrival rate  $\lambda$  and service rate  $\mu$ . The state of the Markov chain is the number of customers in the queue or in service.

In particular, it follows that this invariant distribution is such that

$$\pi(i)q(i, j) = \pi(j)q(j, i), \text{ for all } i, j \geq 0.$$

For instance,  $\pi(n)q(n, n+1) = (1 - \rho)\rho^n\lambda = (1 - \rho)\rho^{n+1}\mu = \pi(n+1)q(n+1, n)$ . It follows from Theorem 9.1.8 that the rate matrix  $Q$  is also the rate matrix of the Markov chain  $x$  reversed in time. Thus, the Markov chain has the same rate matrix when it is reversed in time, and it is therefore statistically the same after time reversal. We say that the Markov chain is *time reversible*.

Note that the departures from the M/M/1 queue before time  $t$  become the arrivals after time  $t$  when the time is reversed. Since the queue remains an M/M/1 queue after time reversal, the arrivals after time  $t$  are a Poisson process with rate  $\lambda$  that is independent of the queue length  $x_t$  at time  $t$ . We then reach the following conclusion.

**Theorem 9.3.1** Consider a stationary M/M/1 queue. The departures before time  $t$  from that queue are a Poisson process with rate  $\lambda$  that is independent of the state of the queue at time  $t$ .

A queue with that property is said to be *quasi reversible*.

From the invariant distribution of the queue length, we can derive the average queue length and the average delay through the queue. For the average queue length we find

$$E(x_t) = \sum_{n=0}^{\infty} n\pi(n) = \sum_{n=0}^{\infty} n(1 - \rho)\rho^n = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}.$$

For the average delay through the queue, we first perform a direct calculation. The probability that a customer arrives when there are  $n$  customers in the queue is  $\pi(n)$ . To see this, note that

$$P[x_t = n | \text{a customer arrives in}(t, t + \epsilon)]$$

$$= \frac{P[\text{a customer arrives in}(t, t + \epsilon) | x_t = n]P(x_t = n)}{P(\text{a customer arrives in}(t, t + \epsilon))} = \frac{\lambda\epsilon\pi(n)}{\lambda\epsilon} = \pi(n),$$

as claimed. Now, the average delay of a customer who arrives when there are  $n$  customers in the queue is the average of  $n + 1$  service times, that is,  $(n + 1)/\mu$ . Indeed, such a customer must wait for the  $n$  customers who were ahead of him to be served and then for his own service time before he can leave the queue. Consequently, the average delay  $T$  of a customer in the M/M/1 queue is given by

$$T = \sum_{n=0}^{\infty} \frac{n+1}{\mu} \pi(n) = \sum_{n=0}^{\infty} \frac{n+1}{\mu} (1-\rho)\rho^n = \frac{1}{\mu - \lambda}.$$

Note that the average queue length  $L := E(x_t)$  and the average delay  $T$  are related by the relationship

$$L = \lambda T.$$

This relationship, called *Little's result*, holds for very general queuing systems. One intuitive justification for Little's result is as follows. Assume that, on average,  $\lambda$  customers go through some service system per unit of time, each customer spends  $T$  units of time in the system, and  $L$  customers are in the system at any time. We want to argue that  $L = \lambda T$ . If each customer pays the system at a unit rate while in the system, the system gets paid at an average rate equal to the average number  $L$  of customers in the system. On the other hand, each customer pays the system an average amount  $T$  equal to the average time spent in the system. Since  $\lambda$  customers go through the system per unit of time and each pays an average of  $T$ , we conclude that the system gets paid at an average rate equal to  $\lambda T$ . Hence,  $L = \lambda T$ , as we claimed.

The M/M/1 queue models a transmitter with rate  $c$  bps where packets arrive as a Poisson process with rate  $\lambda$  and have independent, identically distributed (iid) lengths exponentially distributed with rate  $\mu c$  bits. Indeed, the successive transmission times of the packets are then exponentially distributed with rate  $\mu$ . The results of this section enable us to calculate the average delay of the packets going through the transmitter. Note that if the utilization  $\rho$  of the transmitter does not exceed 80%, then the average delay  $T$  per packet does not exceed  $5/\mu$ . That is, the average delay does not exceed five packet transmission times. (The average packet transmission time is equal to  $\mu^{-1}$ .)

In many communication systems, the interarrival times of packets and the service times are not exponentially distributed. We introduce a model for deterministic service times (such as in ATM networks) with batch arrivals.

### 9.3.2 Discrete-Time Queue

We model an ATM transmitter with a queue that has constant service times (equal to 1) and where  $A_n$  customers arrive at time  $n$ , for  $n \geq 0$ . We assume that the random variables  $\{A_n, n \geq 0\}$  are iid, with mean  $\lambda$  and variance  $\sigma^2$ . To analyze the queue we can observe the number of customers in the system at the successive times. Let  $Y_n$  be the number of customers in the queue at time  $n$ . We assume that the queue operates as follows. At the beginning of the  $n$ th time epoch, there are  $Y_n$  customers in the queue. The server then serves one of these customers (if  $Y_n > 0$ ), so that there are  $(Y_n - 1)^+$  customers just after the service completion. The next batch of  $A_n$  customers then enters the queue. Consequently,

$$Y_{n+1} = A_n + (Y_n - 1)^+ = A_n + Y_n - 1\{Y_n > 0\}.$$

Because the  $A_n$  are iid, the process  $\{Y_n, n \geq 1\}$  is a discrete-time Markov chain. If we assume that  $Y_n$  and  $Y_{n+1}$  have the same distribution, that is, that the queue is in steady state, then we can use the above equation to calculate the mean queue length as follows. We first calculate the mean value of both sides of the identity above. We find

$$EY_{n+1} = EA_n + EY_n - P(Y_n > 0).$$

Since  $EY_{n+1} = EY_n$ , we conclude that  $P(Y_n > 0) = EA_n = \lambda$ . Next we take the mean value of the squares of both sides of the identity. When we use the independence of  $A_n$  and  $Y_n$  and the identity  $Y_n 1(Y_n > 0) = Y_n$ , we find

$$\begin{aligned} E(Y_{n+1})^2 &= E(A_n + Y_n - 1\{Y_n > 0\})^2 \\ &= E(A_n)^2 + E(Y_n)^2 + P(Y_n > 0) \\ &\quad + 2EA_n EY_n - 2EA_n P(Y_n > 0) - 2EY_n \\ &= \sigma^2 + \lambda^2 + E(Y_n)^2 + \lambda \\ &\quad + 2\lambda EY_n - 2\lambda^2 - 2EY_n. \end{aligned}$$

In steady state,  $E(Y_{n+1})^2 = E(Y_n)^2$ , so that we can simplify the last equality and solve for  $EY_n$ . We find

$$EY_n = L = \frac{\sigma^2 - \lambda^2 + \lambda}{2(1 - \lambda)} = \frac{\sigma^2}{2(1 - \lambda)} + \frac{\lambda}{2}. \quad (9.18)$$

The maximum delay experienced by a customer is equal to the maximum backlog of the queue. Let us consider a bound  $D$  on the average delay. Since

$\lambda < 1$ , the above equality shows that the constraint  $L \leq D$  is satisfied if

$$\frac{\sigma^2}{2(1 - \lambda)} \leq D - 0.5,$$

or

$$\lambda + \frac{\sigma^2}{2D - 1} \leq 1. \quad (9.19)$$

As an illustration of this rule, assume that the sequence  $A_n = A_n^1 + \dots + A_n^K$  for  $n \geq 0$ , where for each  $k = 1, \dots, K$  the random variables  $\{A_n^k, n \geq 0\}$  are iid with mean  $\lambda_k$  and variance  $\sigma_k^2$ . Then we find that  $\lambda = \lambda_1 + \dots + \lambda_K$  and  $\sigma^2 = \sigma_1^2 + \dots + \sigma_K^2$ . Consequently, the inequality (9.19) becomes

$$\sum_{k=1}^K \left[ \lambda_k + \frac{\sigma_k^2}{2D - 1} \right] \leq 1.$$

We can write this equation as

$$\sum_{k=1}^K \alpha_k(D) \leq 1 \text{ where } \alpha_k(D) := \lambda_k + \frac{\sigma_k^2}{2D - 1}. \quad (9.20)$$

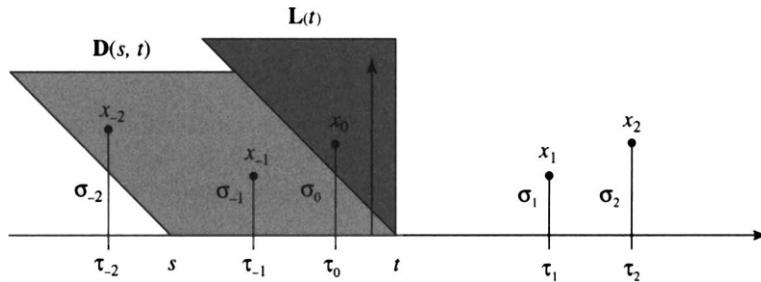
We can think of  $\alpha_k(D)$  as being the *equivalent bandwidth* of source  $k$  and of inequality (9.20) as stating that the sum of the equivalent bandwidths of the sources should not exceed the bandwidth of the transmitter (1 cell per unit of time). The inequality (9.20) indicates that the equivalent bandwidth of a source increases with its variance and decreases with the acceptable average delay through the queue.

### M/GI/ $\infty$ Queue

The M/GI/ $\infty$  queue models independent delays. The customers arrive as a Poisson process with rate  $\lambda$ , and there are infinitely many servers waiting for customers to arrive. As soon as a customer arrives, a server starts serving her. The service times  $\{\sigma_n, n \in \mathbb{Z}\}$  are iid. Thus, if the arrival time of customer  $n$  is  $\tau_n$ , then her departure time is  $\tau_n + \sigma_n$ .

Consider the set of points  $X := \{x_n := (\tau_n, \sigma_n), n \in \mathbb{Z}\}$  in the plane, as shown in Figure 9.6. The points  $x_n$  in the infinite triangle  $L(t)$  defined as

$$L(t) := \{x = (\tau, \sigma) | \tau \leq t \text{ and } \tau + \sigma > t\}$$



9.6

FIGURE

The arrival and service times of customers at an  $M/GI/\infty$  queue. The points in the infinite triangle  $L(t)$  correspond to customers in the queue at time  $t$ . The points in the infinite parallelogram  $D(s, t)$  correspond to customers that leave the queue during  $(s, t]$ .

correspond to customers that are in the queue at time  $t$ . Indeed, if  $\tau_n \leq t$ , then customer  $n$  arrived before time  $t$ . Also, if  $\tau_n + \sigma_n > t$ , then the customer leaves at time  $\tau_n + \sigma_n > t$ .

The points  $x_n$  in the infinite parallelogram  $D(s, t]$  defined as

$$D(s, t) := \{x = (\tau, \sigma) | s < \tau + \sigma \leq t\}$$

correspond to customers that leave the queue during  $(s, t]$ . Indeed,  $\tau_n + \sigma_n$  is the departure time of customer  $n$ .

We claim that the random set of points  $X$  defines a Poisson measure in the plane. That means that the numbers  $N(A_1), \dots, N(A_M)$  of points of  $X$  in disjoint sets  $A_1, \dots, A_M$  are independent random variables that are Poisson distributed with means  $\lambda_1, \dots, \lambda_M$ , where

$$\lambda_m := \lambda \int_{A_m} \int f(\sigma) d\tau d\sigma, m = 1, \dots, M$$

and where  $f(\cdot)$  is the probability density of the service times.

To verify the claim, choose  $\epsilon \ll 1$  and consider  $L_m = (m\epsilon, (m+1)\epsilon] \times [0, \infty)$  for  $m \in \mathbb{Z}$ . The random variables  $N(L_m)$  for  $m \in \mathbb{Z}$  are independent random variables that are Poisson with mean  $\lambda\epsilon$  because they are the increments of the Poisson arrival process with rate  $\lambda$  over intervals of duration  $\epsilon$ . Now consider the sets  $S_{mn} = (m\epsilon, (m+1)\epsilon] \times [n\epsilon, (n+1)\epsilon]$  for  $m \in \mathbb{Z}$  and  $n \geq 0$ . The random variables  $N(S_{mn})$  for  $n \geq 0$  are obtained by sampling the Poisson random variable  $N(L_m)$  with probabilities  $p_n := f(n\epsilon)\epsilon$ . That is, each of the  $N(L_m)$  points of  $X$  in the set  $L_m$  is a point of  $S_{mn}$  with probability  $p_n$ . Indeed, each

customer who arrives during  $[m\epsilon, (m + 1)\epsilon]$  has a service time in the interval  $[n\epsilon, (n + 1)\epsilon]$  with probability  $p_n$ .

It is a simple exercise to verify (see section 9.7, problem 10) that such a sampling of a Poisson random variable produces independent Poisson random variables with mean values  $\lambda\epsilon p_n$ , respectively. Also, by adding independent Poisson random variables, one obtains a Poisson random variable. Thus, by approximating the sets  $A_m$  by unions of squares  $S_{mn}$  and by using the independence and Poisson distribution of the random variables  $N(S_{mn})$ , one concludes that the random variables  $N(A_m)$  are indeed independent Poisson random variables with the mean values indicated in the claim.

In particular, if we choose a collection of times  $t_1 < t_2 < \dots < t_m < t$ , we see by considering the disjoint sets  $D(t_1, t_2), \dots, D(t_m, t)$ , and  $L(t)$  that the departures from the queue during the intervals  $[t_1, t_2], \dots, [t_m, t]$  are independent Poisson random variables with mean values  $\lambda(t_2 - t_1), \dots, \lambda(t - t_m)$ , respectively, and that they are independent of the number  $N(L(t))$  of customers in the queue at time  $t$ .

The mean value of the queue length  $N(L(t))$  is equal to

$$\begin{aligned} L &= \lambda \iint_{L(t)} f(\sigma) d\tau d\sigma \\ &= \lambda \int_0^\infty P(\sigma_1 > t) dt = \lambda E\sigma_1 =: \rho. \end{aligned}$$

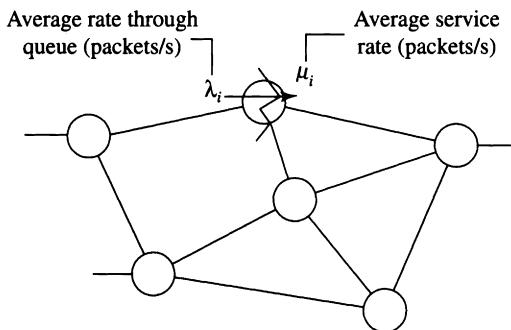
We have proved the following result.

**Theorem 9.3.2** The M/GI/ $\infty$  queue is quasi reversible. That is, the departure process from the queue before time  $t$  is a Poisson process that is independent of the queue length at time  $t$ . Moreover, the queue length is a Poisson random variable with mean  $\rho := \lambda E\sigma_1$ .

### 9.3.3 Jackson Network

We now examine a network of  $J$  M/M/1 queues as shown in Figure 9.7. Customers arrive from outside as independent Poisson processes with rate  $\gamma_i$  into queue  $i$ . When a customer leaves queue  $i$ , he joins queue  $j$  with probability  $r(i, j)$ , independently of the past evolution of the network. On leaving queue  $i$ , a customer leaves the network with probability

$$r(i, 0) := 1 - \sum_{j=1}^J r(i, j).$$



**9.7**  
**FIGURE**

Jackson network. This is a network of single-server queues. The service times in the various queues are independent and are exponentially distributed with rate  $\mu_i$  in queue  $i$ . Customers arrive from outside as independent Poisson processes with rate  $\gamma_i$  into queue  $i$ . The routing is Markov.

The service times of the customers are independent, and they are exponentially distributed with rate  $\mu_i$  in queue  $i$ .

We assume that the probabilities  $r(i, j)$  are such that every customer in the network eventually leaves it. Under this assumption, there is a unique solution to the following flow-conservation equations

$$\lambda_i = \gamma_i + \sum_{j=1}^J \lambda_j r(j, i), \text{ for } i = 1, \dots, J. \quad (9.21)$$

For  $t \geq 0$  define

$$x_t = (x_t^1, \dots, x_t^J),$$

where  $x_t^j$  is the length of queue  $j$  at time  $t$ .

Because of the memoryless property of the exponential distribution and of the way the routing decisions are taken, the process  $\mathbf{x} = \{x_t, t \geq 0\}$  is a Markov chain. The following theorem gives the invariant distribution of that Markov chain.

**Theorem 9.3.3** Assume that the solution  $(\lambda_1, \dots, \lambda_J)$  of (9.21) is such that  $\lambda_i < \mu_i$  for  $i = 1, \dots, J$ . Then the Markov chain  $\mathbf{x}$  admits the following invariant distribution:

$$\pi(x^1, \dots, x^J) = \pi_1(x^1) \cdots \pi_J(x^J), \quad (9.22)$$

where, for  $j = 1, \dots, J$ ,

$$\pi_j(n) = (1 - \rho_j)\rho_j^n, \text{ for } n \geq 0 \text{ where } \rho_j := \frac{\lambda_j}{\mu_j}. \quad (9.23)$$

We give a simple proof of this theorem that uses time reversal. Define a new network with the same M/M/1 queues but with different arrival rates  $\gamma'_j$  and different routing probabilities  $r'(i, j)$ . These values are selected so that

$$\lambda_i r(i, j) = \lambda_j r'(j, i), \text{ for } i, j \in \{1, \dots, J\}$$

and

$$\gamma'_i = \lambda_i r(i, 0).$$

These rates are calculated by *reversing the arrow* in Figure 9.7. Denote by  $Q'$  the rate matrix of this new network. A direct verification shows that

$$\pi(x)q(x, y) = \pi(y)q'(y, x), \text{ for all } x \text{ and } y. \quad (9.24)$$

For instance, let  $y = x + e_j - e_i$  so that a transition from  $x$  to  $y$  occurs when a customer moves from queue  $i$  to queue  $j$ . You then find that

$$q(x, y) = \mu_i r(i, j) \text{ and } q'(y, x) = \mu_j r'(j, i).$$

Equation (9.24) then reads

$$\pi(x)\mu_i r(i, j) = \pi(x + e_j - e_i)\mu_j r'(j, i),$$

and this equation is satisfied in view of the form of  $\pi$  and of the definition of  $r'(j, i)$ . Theorem 9.1.8 then implies that  $\pi$  is indeed the invariant distribution for the Markov chain  $\mathbf{x}$  as we wanted to prove.

It follows from the above result that the distribution of the vector of queue lengths in the network is identically the same as it would be if the arrival processes at all the queues were independent with rate  $\lambda_i$  at queue  $i$ . (It can be shown that the processes are not Poisson in general and that they are certainly not independent.)

In particular, we conclude that the average queue length in queue  $i$  is the same as the average queue length of an M/M/1 queue with arrival rate  $\lambda_i$  and with service rate  $\mu_i$ . This average queue length is

$$L_i = \frac{\lambda_i}{\mu_i - \lambda_i}.$$

Consequently, the average number  $L$  of customers in the network is given by

$$L = \sum_{i=1}^J L_i = \sum_{i=1}^J \frac{\lambda_i}{\mu_i - \lambda_i}.$$

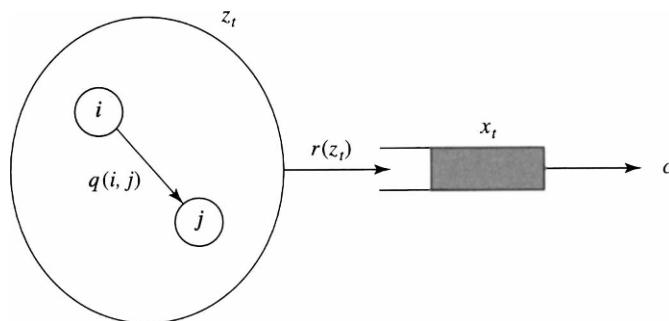
Using Little's result, we conclude that the average delay  $T$  of a customer in the network is given by

$$T = \frac{1}{\gamma} \sum_{i=1}^J \frac{\lambda_i}{\mu_i - \lambda_i}, \quad (9.25)$$

where  $\gamma = \sum_{j=1}^J \gamma_j$  is the total rate at which customers enter the network.

### 9.3.4 Buffer Occupancy for an MMF Source

In section 8.4.3 we described a stochastic source as a Markov-modulated fluid (MMF). We considered the situation in which this source feeds into a buffer that is drained at a constant rate. We derive the stationary distribution of the buffer occupancy. The state of the Markov source is  $z_t$ , the occupancy of the buffer is  $x_t$ . When  $z_t = j$ , the source emits fluid at rate  $r(j)$ . The transmission rate is  $c$ . When the source is in state  $j$ , the buffer fills up at rate  $r(j) - c$ , so that the derivative with respect to  $t$  of  $x_t$  is  $r(j) - c$  when  $x_t > 0$ . (See Figure 9.8.)



9.8

FIGURE

When the source is in state  $z_t$ , it emits fluid at rate  $r(z_t)$ . The buffer is drained at rate  $c$ .

Define the steady-state distribution of the Markov chain  $(z_t, x_t)$  by

$$\pi(i, x) = P(z_t = i \text{ and } x_t \leq x).$$

Consider the evolution of the buffer occupancy and of the source between time  $t$  and time  $t + dt$ . Note that, for  $x > 0$ ,  $z_{t+dt} = i$  and  $x_{t+dt} \leq x$  if and only if for some  $j$  one has  $z_t = j$ ,  $x_t \leq x + (c - r(j))dt$ , and if the source then jumps from state  $j$  to state  $i$ . Indeed, under these conditions, during most of the interval  $[t, t + dt]$  the source produces fluid at rate  $r(j)$  so that the buffer occupancy increases by  $(r(j) - c)dt$ .

Consequently, for  $x > 0$ ,

$$\begin{aligned}\pi(i, x) &= \sum_{j \neq i} \pi(j, x + (c - r(j))dt) q(j, i) dt + \pi(i, x + (c - r(i))dt) (1 + q(i, i)dt) \\ &= \sum_{j \neq i} [\pi(j, x) q(j, i) dt] + \pi(i, x) (1 + q(i, i)dt) + \frac{\partial}{\partial x} \pi(i, x) (c - r(i)) dt.\end{aligned}$$

Hence,

$$\frac{\partial}{\partial x} \pi(i, x) = -(c - r(i))^{-1} \sum_j \pi(j, x) q(j, i).$$

Denote by  $\pi(x)$  the row vector

$$\pi(x) = [\pi(1, x), \dots, \pi(M, x)]$$

and by  $\frac{d}{dx} \pi(x)$  the row vector

$$\frac{d}{dx} \pi(x) = \left[ \frac{\partial}{\partial x} \pi(1, x), \dots, \frac{\partial}{\partial x} \pi(M, x) \right].$$

We can then rewrite the differential equations as

$$\frac{d}{dx} \pi(x) = \pi(x) A$$

where  $A$  is the matrix  $A = [a(i, j), 1 \leq i, j \leq M]$  with

$$a(i, j) = \frac{q(i, j)}{r(j) - c}.$$

To solve these equations we must take into account the boundary conditions that we derive by considering the case  $x = 0$ . Define the set  $S$  of possible values

$j$  of  $z$  such that  $r(j) \leq c$ . By reproducing the derivation above for  $x = 0$  we find that for  $i \in S$  one has

$$\frac{\partial}{\partial x} \pi(i, 0) \{r(i) - c\} = \sum_{j \in S} \pi(j, 0) q(j, i).$$

Similarly, we find that for  $i$  not in  $S$ ,  $\pi(i, 0) = 0$ .

These boundary equations enable us to solve the linear equations. We can write the solution as follows:

$$\pi(x) = \sum_{k=0}^{M-1} a(k) e^{\beta_k x}, x \geq 0,$$

where the  $\beta_k$  are the eigenvalues of  $A$ —that we assume distinct for simplicity—and the  $a(k)$  are proportional to the corresponding eigenvectors. One eigenvalue, say  $\beta_0$ , is zero and corresponds to the eigenvector  $\phi$  equal to the stationary distribution of  $z_t$ ; the other eigenvalues are negative if the system is positive recurrent. Since  $\pi(i, \infty) = \phi(i)$  we conclude that  $a(0) = \phi$ . That is,

$$\pi(x) = \phi + \sum_{k=1}^{M-1} a(k) e^{\beta_k x}, x \geq 0.$$

To clarify the above calculations, let us solve explicitly for  $\pi(i, x)$  when the MMF is an on-off Markov fluid with on-rate equal to 1. In that case, the process  $z_t$  takes the values 0 and 1 and its rate matrix is such that  $q(0, 1) = \lambda$  and  $q(1, 0) = \mu$ . We have  $r(0) = 0$  and  $r(1) = 1$ , and we assume that  $0 < c < r(1) = 1$ ; otherwise there is no queuing possible. Moreover, we must assume that the average input rate is less than  $c$ , that is,  $\lambda/(\lambda + \mu) < c$ , otherwise the queue fills up and has no invariant distribution. We find that

$$A = \begin{bmatrix} \frac{\lambda}{c} & \frac{\lambda}{1-c} \\ -\frac{\mu}{c} & \frac{-\mu}{1-c} \end{bmatrix}.$$

The eigenvalues of  $A$  are 0 and  $\beta := \lambda/c - \mu/(1 - c) < 0$  with the corresponding eigenvectors  $\phi = [\mu/(\lambda + \mu), \lambda/(\lambda + \mu)]$  and  $v := [1 - c, c]$ .

Thus we know that  $\pi(x) = \phi + ave^{\beta x}$ , and we find the constant  $a$  by using the boundary condition  $\pi(1, 0) = 0$ . This gives  $a = -(\lambda/c)/(\lambda + \mu)$ . Putting all this together yields

$$\pi(0, x) = \phi(0) - \phi(1) \frac{1-c}{c} \exp\{\beta x\}$$

$$\pi(1, x) = \phi(1) - \phi(1) \exp\{\beta x\}$$

with  $\phi(0) = \mu/(\lambda + \mu) = 1 - \phi(1)$  and  $\beta := \lambda/c - \mu/(1 - c)$ . From these expressions we can calculate the steady-state probability that the buffer occupancy exceeds  $x$ . We find

$$P(x_t > x) = 1 - \pi(0, x) - \pi(1, x) = \frac{\phi(1)}{c} \exp\{\beta x\}.$$

Figure 9.9 shows a few examples.

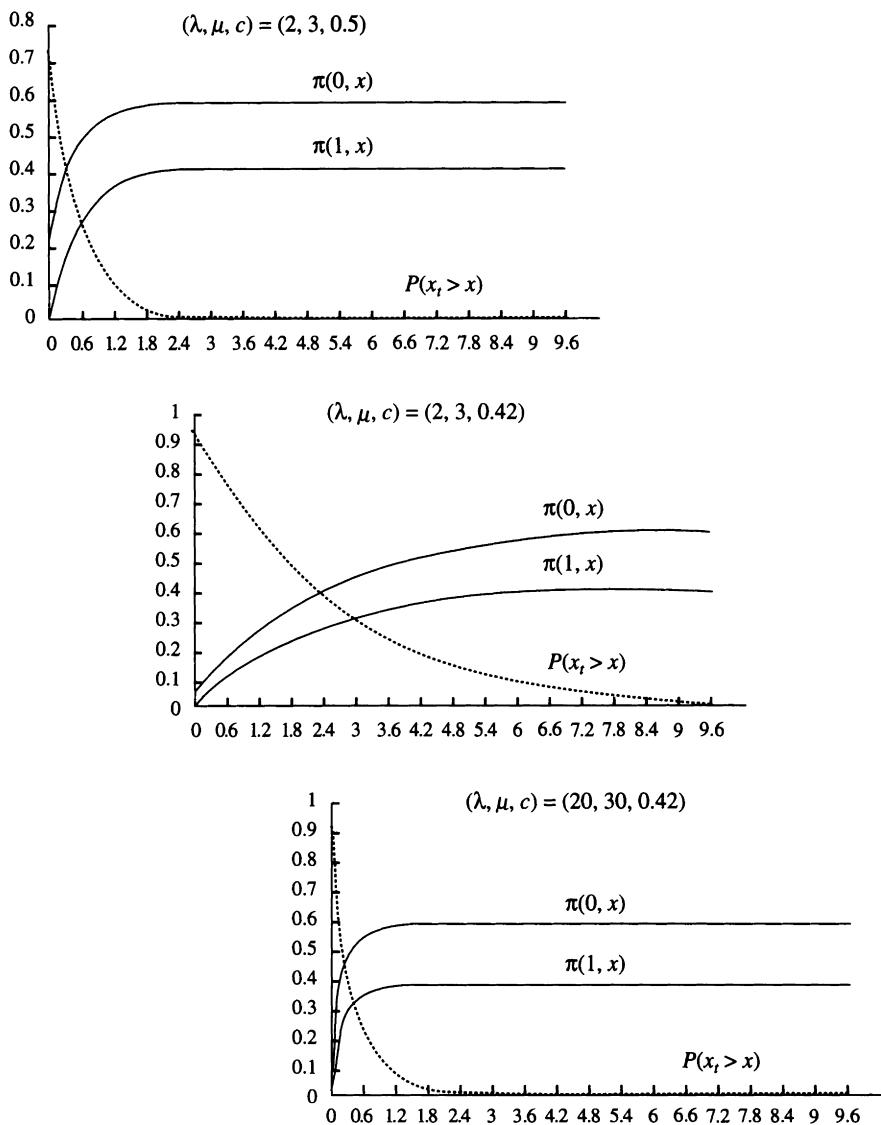
### 9.3.5 Insensitivity of Blocking Probability

We mentioned at the end of section 9.2.1 that the blocking probability for a loss network does not depend on the distribution of the holding times of telephone calls. We now prove that result.

Consider the network in the left side of Figure 9.10 (page 464). The figure shows a closed network with one  $M/GI/\infty$  queue and one  $M/M/1$  queue with service rate  $\lambda$ . Assume that there are  $N$  customers in the network. The outputs of the  $M/M/1$  queue are calls being placed, and the outputs of the  $M/GI/\infty$  queue are calls being terminated. Calls are placed with rate  $\lambda$  as long as there are fewer than  $N$  calls in progress (represented by all  $N$  customers in the  $M/GI/\infty$  queue). When there are  $N$  calls in progress, new calls are blocked. Thus, the call blocking probability is the probability that there are  $N$  customers in the  $M/GI/\infty$  queue. We show that this probability depends on the distribution of the call holding times—the service times in the  $M/GI/\infty$  queue—only through their mean value.

To analyze the network, we introduce arbitrarily small delays of duration  $\epsilon \ll 1$  as shown in the right-hand side of Figure 9.10. Assume that at time 0, the states of the  $M/GI/\infty$  queue and the  $M/M/1$  queue are independent and have the invariant distributions that correspond to Poisson inputs with rate  $\alpha < \lambda$ . Moreover, assume that the states of the little delay lines at time 0 are such that their outputs during  $[0, \epsilon]$  are independent Poisson processes with rate  $\alpha > 0$  and that these outputs are independent of the states of the queues. Consequently, during  $[0, \epsilon]$  the *inputs* of the two queues are independent Poisson processes with rate  $\alpha$ .

Moreover, because the queues have the invariant distribution for those input processes and are quasi reversible, the *outputs* of these queues are independent Poisson processes during  $[0, \epsilon]$ . In addition, these outputs are independent of the states of the queues at time  $\epsilon$ . Thus, at time  $\epsilon$ , the state of the network has the same distribution as at time 0 and we can repeat the argument during

**9.9****FIGURE**

The invariant distribution of an on-off Markov fluid that feeds a buffer with a constant service rate. Three sets of parameter values are illustrated.



**FIGURE**  
9.10

Network model of loss system. The network in the left-hand side of the figure models a loss system. A customer who enters the  $M/GI/\infty$  queue represents a call being placed. A customer who leaves that queue represents a call being terminated. Thus, calls are placed with rate  $\lambda$  and have iid holding times. The network on the right-hand side is modified by inserting two small delays to simplify the analysis.

$[\epsilon, 2\epsilon]$ , and so on. Consequently, during any time interval  $[n\epsilon, (n+1)\epsilon)$  the outputs of the queues are independent Poisson processes with rate  $\alpha$ .

Moreover, at any time  $t$  the queue lengths  $X$  in the  $M/GI/\infty$  queue and  $Y$  in the  $M/M/1$  queue are independent and are Poisson with mean  $\alpha E\sigma$  and geometric with parameter  $\alpha/\lambda$ , respectively. That is,

$$P(X = m, Y = n) = \phi(m, n) := \frac{(\alpha E\sigma)^m}{m!} e^{-\alpha E\sigma} \left(\frac{\alpha}{\lambda}\right)^n \left(1 - \frac{\alpha}{\lambda}\right), \quad m, n \geq 0. \quad (9.26)$$

We conclude that the above distribution is invariant for the network with the small delays  $\epsilon$ .

Now consider what happens when  $\epsilon \downarrow 0$ . Take an arbitrary realization of the initial state of the network. For  $\epsilon$  small enough, the delay lines are empty. By decreasing  $\epsilon$  further, we modify the arrival times into the queue by  $\epsilon$ . Eventually, for  $\epsilon$  small enough, the lengths of the two queues  $(X_t^\epsilon, Y_t^\epsilon)$  at time  $t$  stop changing as  $\epsilon$  keeps on decreasing. Moreover, these queue lengths are equal to the value they would have for  $\epsilon = 0$  whenever time  $t$  is not an arrival time into one of the queues when  $\epsilon = 0$ . Hence,

$$P((X_t^\epsilon, Y_t^\epsilon) = (m, n)) \rightarrow P((X_t^0, Y_t^0) = (m, n)) \text{ as } \epsilon \downarrow 0.$$

It follows that the distribution (9.26) is invariant for the network without the delay lines.

Note that for this invariant distribution the total number of customers in the network is random. If we know that the total number is  $N$ , then we can derive the invariant distribution of the network as follows. Assume that we have some Markov chain approximation  $\mathbf{x} = \{x_t, t \geq 0\}$  of the state of the network. Such an approximation can be constructed by approximating the holding times by first

passage times of finite Markov chains. Denote by  $\pi$  the invariant distribution of the Markov chain  $x$  on its state space  $X$ . Assume that there is a subset  $Y$  of  $X$  such that the Markov chain  $x$  is irreducible on  $Y$  and cannot leave or enter  $Y$ . That is, the rate matrix  $Q$  of  $x$  is such that

$$q(x, y) = 0 \text{ if } (x, y) \in Y^c \times Y \text{ or } (x, y) \in Y \times Y^c,$$

where  $Y^c := X - Y$ . The claim is then that the distribution  $\pi_Y$  is invariant for  $x$  where

$$\pi_Y(x) := \frac{\pi(x)1\{x \in Y\}}{\pi(Y)}. \quad (9.27)$$

To verify this claim, note that

$$\sum_{x \in Y} \pi(x)q(x, y) = \sum_{x \in X} \pi(x)q(x, y) = 0.$$

The first equality follows from (9.27) and the second from the invariance of  $\pi$ .

Thus, the invariant distribution of the network of the left-hand side of Figure 9.10 is the distribution (9.26) normalized on the set of states that correspond to  $N$  customers in the network. In particular, we find that the invariant probability that there are  $m$  customers in the  $M/GI/\infty$  queue and  $N - m$  in the  $M/M/1$  queue is given by

$$\begin{aligned} P[X = m, Y = N - m | X + Y = N] &= \frac{\phi(m, N - m)}{\sum_{n=0}^N \phi(n, N - n)} \\ &= \frac{\rho^m / m!}{\sum_{n=0}^N \rho^n / n!}, \text{ for } m = 0, \dots, N, \end{aligned}$$

where  $\rho := \lambda E\sigma$ .

In particular, for  $m = N$ , this formula gives the blocking probability that is thus seen to depend on the distribution of the service times only through their mean value.

## 9.4

## ATM NETWORKS

The analysis of ATM networks reflects one important characteristic of these networks: their large bandwidth-delay product. As we explained in Chapter 2, this large bandwidth-delay product makes feedback control largely ineffective. Consequently, ATM networks use open-loop control strategies. In this section

we discuss some basic results that network researchers use to analyze the performance of ATM networks.

In section 9.4.1 we discuss deterministic models of networks. These models enable researchers to gain some insight into the behavior of buffers and leaky buckets. In section 9.4.2 we explain large deviations of iid random variables. We explore a generalization in section 9.4.3. We then apply the results to the analysis of loss probabilities in a buffer in section 9.4.4.

### 9.4.1 Deterministic Approaches

In this section we develop performance bounds for buffers and traffic rates using deterministic rather than stochastic analysis.

#### *Linear Bounds*

Consider a source that transmits at most  $B + Rt$  bits in any interval of  $t$  seconds, for *any* possible value of  $t$ . We say that such a source produces a  $(B, R)$ -traffic to recall these constraints.

Assume that a  $(B, R)$ -traffic goes through a first-come, first-served buffer (initially empty) with a capacity to store  $B$  bits and equipped with a transmitter with rate  $C$  bps, with  $C \geq R$ . We claim that the buffer never loses any bit and that it delays the input stream by at most  $B/C$  seconds.

To verify the claim, assume that the buffer loses bits at some time  $T$  and that it was empty for the last time before  $T$  at time  $T - S$ . Then, during  $[T - S, T]$ , the buffer transmits exactly  $CS$  bits and at most  $B + RS$  bits enter the buffer. Since  $B + RS - CS \leq B$ , it is not possible for the buffer occupancy at time  $T$  to exceed  $B$ . This contradicts the assumption that the buffer loses bits at time  $T$ .

Consider a node that transmits bits from its buffer at rate  $C$  whenever the buffer is nonempty. Assume that one  $(B, R)$ -traffic and another  $(B', R')$ -traffic share that buffer. Assume that  $C > R + R'$ . The input stream is a  $(B + B', R + R')$ -traffic. If the node transmits the bits in their order of arrival, then the node delays the  $(B, R)$ -traffic by at most  $(B + B')/C$ . However, if we do not assume that the node sends the bits in their order of arrival but only that the node keeps transmitting whenever it is nonempty, then we find that the node delays its input traffic streams by at most  $T = (B + B')/(C - R - R')$ .

To see this, note that the buffer cannot remain nonempty for an interval with a duration longer than  $T$ , since in that interval the node transmits  $CT$  bits, and at most  $B + B' + (R + R')T = CT$  bits enter the buffer. Moreover, if the node transmits all the bits of the  $(B, R)$ -stream in their order of arrival, then the node delays these bits by at most  $S = (B + B')/(C - R')$ . Indeed, the worst

backlog facing a bit of the  $(B, R)$ -traffic is  $B + B'$  and the node runs out of that backlog and of the subsequent arrivals of the  $(B', R')$ -traffic after  $S$  seconds.

Let us summarize the above observations in the form of a theorem.

**Theorem 9.4.1** By definition, a  $(B, R)$ -traffic carries at most  $B + Rt$  bits in any time interval of duration  $t$  seconds. A network node can avoid losing bits of a  $(B, R)$ -traffic by reserving a buffer capacity of  $B$  bits and a bandwidth of  $R$  bps for that traffic.

Moreover, a network node that transmits at rate  $C$  whenever it is not empty and that is shared by a  $(B, R)$ -traffic and a  $(B', R')$ -traffic delays its inputs by at most  $(B + B')/(C - R - R')$  s. If the node transmits the bits of the  $(B, R)$ -traffic in their order of arrival, then it delays those bits by at most  $(B + B')/(C - R')$  s.

This theorem can be used to derive bounds on delays in network nodes and therefore the end-to-end network delay.

### *Leaky Bucket*

To ensure that the source satisfies the above conditions, the user can control that source with a  $(B, R)$ -leaky bucket. This leaky bucket has a counter that accumulates credits (tokens) at the continuous rate  $R$  and can accumulate  $B$  units of credit. To transmit a bit, the counter must contain at least one unit of credit.

Let us verify that the output of the  $(B, R)$ -leaky-bucket controller is a  $(B, R)$ -traffic. To do this, we consider an interval of time  $[S, S + t]$  of duration  $t$  ( $t \geq 0$ ). We must show that the output of the leaky-bucket controller produces at most  $B + Rt$  bits in that time interval. Let  $A$  be the number of tokens in the counter at time  $S$ . Note that  $A \leq B$ . During the time interval  $[S, S + t]$  the counter accumulates  $Rt$  tokens. Thus,  $A + Rt$  tokens are available to the source to transmit bits. Consequently, the output produces at most  $A + Rt \leq B + Rt$  bits in the time interval of duration  $t$ . Hence the output is indeed a  $(B, R)$ -traffic, as claimed.

We have seen that it is a simple matter for the user to guarantee that the traffic is a  $(B, R)$ -traffic. What is more difficult to evaluate is the effect of the leaky-bucket controller on the source traffic. The leaky-bucket controller delays the bit stream that the source produces. The statistics of the delay depend on the statistics of the source bit stream.

Thus, by using a leaky-bucket controller and by reserving buffer capacity and bandwidth, the network can guarantee that it will not lose user bits and that it will not introduce a delay larger than  $B/R$ . However, it is up to the user to select the parameters  $(B, R)$  so that the controller does not delay the source traffic excessively.

The network can verify that the traffic is a  $(B, R)$ -traffic by checking whether the traffic goes through a  $(B, R)$ -leaky-bucket controller without any delay. Indeed, only a  $(B, R)$ -traffic can go without delay through a  $(B, R)$ -leaky-bucket controller.

Assume that a user sets up a connection and specifies that the traffic will be a  $(B, R)$ -traffic. What should the network do if the traffic does not go through the  $(B, R)$ -controller? The ATM Forum recommendation is that the network should mark the cells that the leaky bucket delays as low-priority cells by setting their CLP header bit. These cells become candidates for discarding by network nodes that experience congestion.

We now show that the  $(B, R)$ -leaky-bucket controller—we call it  $LB$ —delays the traffic the least among all the first in, first out controllers that output a  $(B, R)$ -traffic. Consider the leaky-bucket controller  $LB$  and another controller  $LB'$  whose output is also a  $(B, R)$ -traffic. The two controllers have the same input. We claim that every bit leaves  $LB$  before  $LB'$ . To prove the claim we argue by contradiction. Assume that at least one bit leaves  $LB'$  before  $LB$  and that this occurs for the first time at time  $t$ . Thus, at time  $t$  the token counter of  $LB$  must be empty. Denote by  $t - T$  the last time before time  $t$  that the token counter of  $LB$  was full. It follows that during  $[t - T, t]$  the output of  $LB$  carries  $B + R \times T$  bits. Moreover, the output of  $LB'$  during  $[t - T, t]$  must carry at least the same bits as the output of  $LB$  plus one more. Indeed, before time  $t$ , all the bits leave  $LB'$  after  $LB$ , and the output of  $LB'$  catches up with and exceeds that of  $LB$  at time  $t$ . But this fact implies that  $LB'$  outputs at least  $B + R \times T + 1$  bits in  $T$  seconds. Consequently, the output of  $LB'$  is not a  $(B, R)$ -traffic.

As we explained earlier, the GCRA differs slightly from the leaky bucket described above. The difference is the quantization: the  $\text{GCRA}(T, \tau)$  is a leaky bucket that removes  $T$  units of fluid per cell and not one infinitesimal unit per bit. We analyzed in section 8.4.2 the effect of a GCRA controller on the traffic.

### Burstiness

We can use deterministic models to clarify the notion of burstiness. Roughly speaking, a traffic stream is more bursty than another if it requires more buffering at a transmitter. After defining this notion more precisely, we show that a leaky-bucket controller reduces the burstiness.

For the purpose of this discussion, it is easier to view traffic as a fluid than as a stream of discrete packets. We define a message as a time-varying bit rate. That is, a message  $m$  is a nonnegative function of time  $m = \{m(t), t \geq 0\}$ . We assume that the function  $m$  is integrable and that  $\int_0^\infty m(t)dt = M$ . The interpretation is that  $m(t)$  is the bit rate of the message at time  $t$  and that  $M$  is the total number of bits of the message.

Assume that  $\mathbf{m}$  goes through a buffer with service rate  $c$  bps. We denote the maximum number of bits that the buffer must store by  $b_m(c)$ . To calculate  $b_m(c)$ , we note that the buffer occupancy  $x(t)$  satisfies, with  $x(0) = 0$ ,

$$\frac{d}{dt}x(t) = \begin{cases} m(t) - c, & \text{if } x(t) > 0 \\ (m(t) - c)^+, & \text{if } x(t) = 0. \end{cases}$$

These equations express that the buffer occupancy grows at a rate equal to the input rate  $m(t)$  minus the service rate  $c$  and that the buffer occupancy cannot become negative. By solving the equations, we can find the maximum value  $b_m(c)$  of  $x(t)$  for  $t \geq 0$ .

We can define a  $(B, R)$ -leaky-bucket controller as a device that accumulates a token fluid at a constant rate  $R$  in a token buffer that can store up to  $B$  units of token fluid. To transmit  $\epsilon$  units of (traffic) fluid, the transmitter must remove  $\epsilon$  units of token fluid.

Consider some message  $\mathbf{m}$ . Assume that this message goes through a  $(B, R)$ -leaky-bucket controller. The output is a new message  $\mathbf{n}$ . The claim is that  $b_n(c) \leq b_m(c)$  for all  $c \geq 0$ . That is, the output of the leaky bucket requires less buffering at a transmitter of any fixed rate  $c$ . Note that this buffering at the transmitter does not include the buffering in the leaky-bucket controller. We say that the message  $\mathbf{n}$  is *less bursty* than message  $\mathbf{m}$ .

To verify the claim, consider the situation where the message  $\mathbf{n}$  is served by a buffer with a transmitter with rate  $c$ . Assume that the buffer accumulates  $b$  units of fluid. We will show that the buffer also accumulates at least  $b$  units of fluid when its input is  $\mathbf{m}$ . To show this, we denote by  $T$  the first time that the buffer occupancy reaches the value  $b$  with the input  $\mathbf{n}$  and by  $S$  the last time before  $T$  that the buffer was empty. Thus, during  $[S, T]$  the input  $\mathbf{n}$  must carry  $b + c(T - S)$  units of fluid.

Let  $K$  be the amount of fluid accumulated in the leaky bucket controller at time  $S$ . If  $K = 0$ , then  $\mathbf{m}$  carries at least  $b + c(T - S)$  units in  $[S, T]$  and the claim is proved. If  $K > 0$ , let  $U$  be the last time before  $S$  that the leaky bucket controller is empty. The claim is that during  $[U, T]$ ,  $\mathbf{m}$  carries at least  $b + c(T - U)$  units of fluid. To see this, note that to accumulate  $K$  units during  $[U, S]$ ,  $\mathbf{m}$  carries at least  $K + R(S - U) > K + c(S - U)$  units of fluid during that time interval. Moreover, during  $[S, T]$ ,  $\mathbf{m}$  must carry at least  $b + c(T - S) - K$  units, otherwise  $\mathbf{n}$  could not carry  $b + c(T - S)$  units during  $[S, T]$ . This completes the proof.

Some researchers have proposed the following variation on the linear bounds. Instead of defining a  $(B, R)$ -traffic, they define a  $\{(B_1, R_1), \dots, (B_K, R_K)\}$ -traffic as a stream that carries at most  $B_k + R_k t$  cells for all  $k = 1, \dots, K$  and for all  $t \geq 0$ . To enforce that condition, the source traffic should go through each of the  $(B_k, R_k)$ -regulators for  $k = 1, \dots, K$ . For instance, we may recall that the VBR specification calls for two leaky buckets; see section 8.4.2.

### 9.4.2 Large Deviations of iid Random Variables

In this section we explore statistical methods for call admission. We want to justify the effective bandwidth results. We first explain the case of iid random variables. In the next section we extend the results to nonindependent random variables.

#### *Cramer's Theorem*

Consider a collection  $\{X_n, n \geq 1\}$  of independent and identically distributed random variables with common distribution  $F(\cdot)$  and with finite mean value  $m$ . Define partial sum  $S_n$  as

$$S_n = \sum_{k=1}^n X_k.$$

We know from the strong law of large numbers that

$$\frac{S_n}{n} \rightarrow m \text{ as } n \rightarrow \infty \text{ with probability 1.}$$

Thus, the probability that  $S_n/n$  is away from  $m$  goes to 0 as  $n$  increases. It can be shown that this convergence to 0 occurs exponentially fast in  $n$ . More precisely, for  $a \geq m$ ,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log P(S_n \geq na) = -\Lambda^*(a), \quad (9.28)$$

where

$$\Lambda^*(a) = \sup_{\theta} [\theta a - \Lambda(\theta)] \text{ with } \Lambda(\theta) = \log E(e^{\theta X_1}). \quad (9.29)$$

For this result to be valid we need  $\Lambda(\theta)$  to be differentiable and finite in a neighborhood of 0. Algebra shows that  $\Lambda^*(m) = 0$ .

Roughly, this result states that

$$P(S_n \approx na) \approx e^{-n\Lambda^*(a)}.$$

This result is called Cramer's Theorem.

That is, it is unlikely that  $S_n/n$  is away from  $m$ , and this is exponentially unlikely in  $n$ . The value of  $\Lambda^*(a)$  indicates how difficult it is for  $S_n/n$  to be close to  $a$ . If  $\Lambda^*(a)$  is large, then it is very difficult for  $S_n/n$  to be close to  $a$ .

For further reference, we note that the function  $\Lambda(\theta)$  can be obtained from the function  $\Lambda^*(a)$  by

$$\Lambda(\theta) = \sup_a [\theta a - \Lambda^*(a)]. \quad (9.30)$$

One says that  $\Lambda(\cdot)$  and  $\Lambda^*(\cdot)$  are the convex dual of each other.

### *Comments and Sharpening*

Why should we expect this probability to decay exponentially in  $n$ ? Assume that

$$P(S_n \approx na) \approx \epsilon.$$

Then,  $P(X_1 + \dots + X_n \approx na) \approx \epsilon$ . Moreover, the most likely way for  $X_1 + \dots + X_{2n}$  to be close to  $2na$  is for  $X_1 + \dots + X_n$  to be close to  $na$  and for  $X_{n+1} + \dots + X_{2n}$  also to be close to  $na$ . If we believe this last sentence, then

$$\begin{aligned} P(S_{2n} \approx 2na) &= P(X_1 + \dots + X_{2n} \approx 2na) \\ &\approx P(X_1 + \dots + X_n \approx na)P(X_{n+1} + \dots + X_{2n} \approx na) \\ &\approx \epsilon^2, \end{aligned}$$

which shows that  $P(S_n \approx na)$  is approximately exponential in  $n$ . One could argue that there are many ways for  $S_{2n}$  to be close to  $2na$  other than for both  $X_1 + \dots + X_n$  and  $X_{n+1} + \dots + X_{2n}$  to be close to  $na$ . For instance,  $X_1 + \dots + X_n$  could be approximately  $n(a + q)$  for some small  $q$ , and  $X_{n+1} + \dots + X_{2n}$  could be approximately  $n(a - q)$ . The probability of this event is approximately

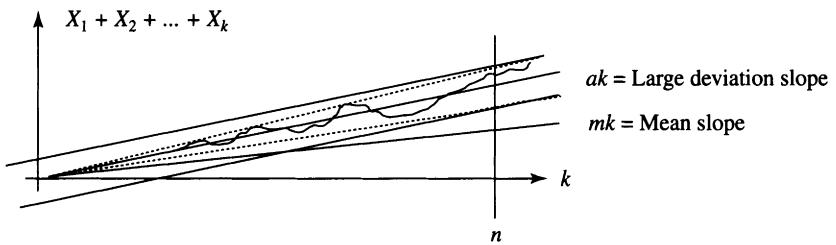
$$e^{-n\Lambda^*(a+q)} \times e^{-n\Lambda^*(a-q)},$$

However,

$$\Lambda^*(a+q) + \Lambda^*(a-q) > 2\Lambda^*(a),$$

by strict convexity of the function  $\Lambda^*(\cdot)$ . Consequently, the probability of that event is much smaller than  $\epsilon^2$ .

It can be shown formally that, given that  $S_n \approx na$ , the most likely realization of that event is if  $S_k \approx ak$  for  $k \leq n$ . Thus, the rare event  $\{S_n \approx na\}$  occurs when the partial sums  $X_1 + \dots + X_k$  are approximately equal to  $ak$  for  $k \leq n$ . We will remember that fact by saying that *the large deviation occurs in a straight line*, as illustrated in Figure 9.11. Thus, the most likely way that the arrival sequence has an empirical rate  $a$  over some long time interval is for the arrivals to occur with an approximately constant rate  $a$  during that interval.



9.11

FIGURE

### *Proof of Cramer's Theorem*

The proof of this theorem illustrates a general approach used in the theory of large deviations. To show a limit, we prove an upper bound and a lower bound. That is, we show

- ♦ for all  $n \geq 1$ ,

$$P(S_n \geq na) \leq \exp\{-n\Lambda^*(a)\}. \quad (9.31)$$

- ♦ for all  $\epsilon, \delta > 0$  there is some  $N$  such that, for all  $n \geq N$ ,

$$P(S_n \geq na - n\epsilon) \geq (1 - \delta) \exp\{-n\Lambda^*(a)\}. \quad (9.32)$$

Let us first show the upper bound (9.31). We use Markov's inequality. For all  $\theta > 0$  one has

$$\begin{aligned} P(S_n \geq na) &\leq E \exp\{\theta(S_n - na)\} = \exp\{-n\theta a\}[E \exp\{\theta X_1\}]^n \\ &= \exp\{-n(\theta a - \Lambda(\theta))\}. \end{aligned}$$

We obtain (9.31) by minimizing the right-hand side of the above inequality over  $\theta > 0$ . (Recall the definition (9.29).)

The lower bound (9.32) is slightly more difficult to prove. The trick is to change the distribution of the  $X_n$ 's to make it easy for them to have a sample mean close to  $a$ . To do this, we define random variables  $Y_n$  that have a distribution such that

$$P(Y_n \in (x, x + dx)) = \exp\{\theta x\}P(X_n \in (x, x + dx))E \exp\{-\theta X\} \quad (9.33)$$

$$= \exp\{-\Lambda(\theta) + \theta x\}P(X_n \in (x, x + dx)). \quad (9.34)$$

The last term in the right-hand side of (9.33) is there to normalize the distribution of  $Y_n$ . The random variables  $Y_n$  for  $n \geq 1$  are iid and we choose  $\theta$  so that  $E(Y_n) = a$ . Note that

$$P\left(\frac{Y_1 + \dots + Y_n}{n} \in [a - \epsilon, a + \epsilon]\right) \rightarrow 1, \text{ as } n \rightarrow \infty,$$

by the weak law of large numbers (because  $E(Y_n) = a$ ). Now we claim that for  $0 < \epsilon \ll 1$

$$\begin{aligned} & P\left(\frac{Y_1 + \dots + Y_n}{n} \in [a - \epsilon, a + \epsilon]\right) \\ &= \exp\{\theta na - n\Lambda(\theta)\}P\left(\frac{X_1 + \dots + X_n}{n} \in [a - \epsilon, a + \epsilon]\right). \end{aligned} \quad (9.35)$$

To see this equality, note that with  $\mathbf{X} = (X_1, \dots, X_n)$  and  $\mathbf{Y} = (Y_1, \dots, Y_n)$  and any function  $f(\cdot)$  one has

$$E\left\{f(\mathbf{X}) \frac{P_{\mathbf{Y}}(\mathbf{X})}{P_{\mathbf{X}}(\mathbf{X})}\right\} = \int f(\mathbf{x}) \frac{P_{\mathbf{Y}}(\mathbf{x})}{P_{\mathbf{X}}(\mathbf{x})} P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \int f(\mathbf{x}) P_{\mathbf{Y}}(\mathbf{x}) d\mathbf{x} = E\{f(\mathbf{Y})\} \quad (9.36)$$

where  $P_{\mathbf{Y}}(\cdot)$  is the probability density of the vector  $\mathbf{Y}$  and similarly for  $P_{\mathbf{X}}(\cdot)$ . We obtain (9.35) by choosing  $f(\mathbf{x}) = 1\{x_1 + \dots + x_n \in [na - n\epsilon, na + n\epsilon]\}$  and by substituting the value of the ratio of densities as specified by (9.33).

Using (9.35) we conclude that for  $n$  large enough

$$\exp\{na\theta - n\Lambda(\theta)\}P\left(\frac{X_1 + \dots + X_n}{n} \geq a - \epsilon\right) \geq (1 - \delta). \quad (9.37)$$

We now claim that  $\theta$  achieves the maximum over  $\sigma$  of  $g(\sigma) := a\sigma - \Lambda(\sigma)$ . This maximum can be shown to be unique because  $\Lambda$  is convex. To prove the claim, we show that  $g'(\theta) = a - \Lambda'(\theta) = 0$ . That is, we show that  $\Lambda'(\theta) = a$ . To derive this identity, we use the fact that

$$a = E(Y_n) = e^{-\Lambda(\theta)} E(X_n e^{\theta X_n}). \quad (9.38)$$

Also, since  $\Lambda$  is differentiable, we can differentiate  $\exp\{\Lambda(\theta)\} = E(\exp\{\theta X_n\})$  to obtain

$$\Lambda'(\theta) e^{\Lambda(\theta)} = E(X_n e^{\theta X_n}). \quad (9.39)$$

Substituting (9.39) into (9.38) proves the claim.

Thus,

$$a\theta - \Lambda(\theta) = \sup_{\sigma} [a\sigma - \Lambda(\sigma)] = \Lambda^*(a).$$

We use the result of this calculation in (9.37) to obtain, for large enough  $n$ ,

$$P\left(\frac{X_1 + \dots + X_n}{n} \geq a - \epsilon\right) \geq (1 - \delta) \exp\{-n\Lambda^*(a)\},$$

which is (9.32).

### 9.4.3 Straight-Line Large Deviations

The results on iid random variables are not sufficient to study communication networks. We need results on sequences of random variables that are not independent. In this section, we define a property of sequences of random variables, and we comment on when that property is satisfied.

#### *Sequences with Straight-Line Large Deviations*

Let  $\{X_n, n \geq 1\}$  be a sequence of random variables. Define

$$S_n := X_1 + \dots + X_n, \text{ for } n \geq 1.$$

Let also  $\Lambda^*$  be a function defined on the real line and taking values in  $[0, \infty]$ . We say that the random variables are of class  $S(\Lambda^*)$  if

$$\begin{aligned} & \lim_{\epsilon \downarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \log P\left(\max_{1 \leq k \leq n} |S_k - ka| \leq n\epsilon\right) \\ &= \lim_{\epsilon \downarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \log P(|S_n - na| \leq n\epsilon) = -\Lambda^*(a) \end{aligned} \quad (9.40)$$

for all  $a \in (-\infty, +\infty)$ .

This property means that the probability that the partial sums  $\{S_k, k \geq 1\}$  follow a straight line with slope  $a$  for  $n$  time units is approximately  $\exp\{-n\Lambda^*(a)\}$ . That is, if the sequence  $S_k$  models some arrival process, then the probability that the sample arrival rate is some value  $a$  larger than the expected rate for some duration  $n$  is given by  $\exp\{-n\Lambda^*(a)\}$ . Moreover, the most likely way that this large sample rate occurs is when the arrival process has a sustained sample rate close to  $a$  over that duration.

For a sequence of class  $S(\Lambda^*)$ , we define

$$\Lambda(\delta) := \sup_a [\delta a - \Lambda^*(a)]. \quad (9.41)$$

Because of (9.30), sequences of iid random variables are of class  $S(\Lambda^*)$ . One can show that functions of finite Markov chains are of class  $S(\Lambda^*)$ . This class of random sequences enables us to model complex traffic streams.

#### 9.4.4 Large Deviation of a Queue

Our objective is to evaluate the loss rate at a queue. That is, the queue has a finite buffer capacity  $B$  and we want to estimate the fraction of arrivals that occur when the queue is full. For many arrival processes, this fraction is comparable to the fraction of times that the queue with an infinite buffer capacity has a buffer occupancy that exceeds  $B$ . That relationship holds for the processes that we consider in this section. Thus, we assume that the queue has an infinite buffer capacity, and we estimate  $P(W > B)$ , the invariant probability that the buffer occupancy  $W$  exceeds  $B$ . Fix  $\delta > 0$ . We show that this probability is such that

$$P(W > B) \approx \exp\{-\delta B\}$$

provided that the service rate of the queue is large enough. For a background on this formulation, see the discussion on buffering in section 8.4.3.

Consider the following discrete-time queuing system. For  $n \geq 1$ ,  $X_n$  customers arrive at the queue at time  $n$ , and up to  $c$  customers that are in the queue are served at that time. The arrival process  $\{X_n\}$  is assumed to be of class  $S(\Lambda^*)$ . Recall that this property means that the most likely way for this process to exhibit a large arrival rate over some interval of time is for it to maintain that large arrival rate during the duration. We try to estimate the probability that, starting empty, the queue occupancy reaches a large value  $B$  before becoming empty again. We argue that this event can happen if the arrivals have an empirical rate  $a > c$  for at least  $B/(a - c)$  time units. According to (9.40), the probability that the arrivals behave in that way is approximately equal to

$$\exp \left\{ -\frac{B}{a - c} \Lambda^*(a) \right\}.$$

Since the rate  $a$  can be any value larger than  $c$ , we find that the probability that the queue occupancy reaches a value  $B$  before becoming empty again is approximately equal to

$$\sum_{a>c} \exp \left\{ -\frac{B}{a - c} \Lambda^*(a) \right\}.$$

We further approximate this sum of exponentials in  $B$  by the exponential with the largest exponent, that is, by

$$\exp \left\{ -\frac{B}{a^* - c} \Lambda^*(a^*) \right\},$$

where

$$\frac{\Lambda^*(a^*)}{a^* - c} = \min_{a>c} \frac{\Lambda^*(a)}{a - c}. \quad (9.42)$$

The argument above shows that the probability that the buffer occupancy reaches a large value  $B$  in a busy cycle decays exponentially in  $B$ , at least in a first approximation. We should note the many approximations that we made along the way. We were only trying to get the rate of decay of the exponential, and we neglected all possible estimates of the coefficient in front of the exponential. For specific models we could calculate the coefficient of the exponential rather precisely. However, since actual traffic models in networks are largely unknown, there is little practical interest in pursuing that line of development.

### *Effective Bandwidth*

A natural question is to ask what the value of  $c$  should be for the decay rate to be equal to a specified value, say  $\delta$ . That is, we want to find the smallest possible value of  $c$  such that

$$\min_{a>c} \frac{\Lambda^*(a)}{a - c} = \delta. \quad (9.43)$$

We denote that value of  $c$  by  $\alpha(\delta)$ , and we call it the *effective bandwidth* of the arrival stream. The interpretation is that the effective bandwidth  $\alpha(\delta)$  is the rate at which that stream must be served so that the buffer occupancy decays as an exponential with rate  $\delta$ .

The claim is that

$$\alpha(\delta) = \frac{\Lambda(\delta)}{\delta}. \quad (9.44)$$

To prove the claim, we note that (9.43) implies that, for all  $c \geq \Lambda(\delta)/\delta$ , one has

$$\delta c \geq \Lambda(\delta) = \sup_a [\delta a - \Lambda^*(a)] = \delta a^* - \Lambda^*(a^*), \text{ for some } a^* > c,$$

so that, for all  $a$ ,

$$\delta c \geq \delta a - \Lambda^*(a)$$

and therefore, for all  $a > c$ ,

$$\frac{\Lambda^*(a)}{a - c} \geq \delta.$$

On the other hand, if  $c < \Lambda(\delta)/\delta$ , then

$$\delta c < \Lambda(\delta) = \delta a^* - \Lambda^*(a^*) \text{ and } \frac{\Lambda^*(a^*)}{a^* - c} < \delta.$$

Consequently,  $\Lambda(\delta)/\delta$  is indeed the smallest value of  $c$  so that  $\min_{a>c}[\Lambda^*(a)/(a - c)] = \delta$ .

If we recall the definition (9.29) of  $\Lambda(\delta)$ , we note the important result that *the effective bandwidth is additive*. That is, if  $X_n = X_n^1 + X_n^2$ , where the sequences of random variables  $\{X_n^1\}$  and  $\{X_n^2\}$  are independent, then we find that

$$\begin{aligned}\alpha(\delta) &= \frac{1}{\delta} \log Ee^{\delta(X_n^1 + X_n^2)} = \frac{1}{\delta} \log [Ee^{\delta X_n^1} \times Ee^{\delta X_n^2}] \\ &= \frac{1}{\delta} \log Ee^{\delta X_n^1} + \frac{1}{\delta} \log Ee^{\delta X_n^2} \\ &= \alpha_1(\delta) + \alpha_2(\delta),\end{aligned}$$

where  $\alpha_i(\delta)$  is the effective bandwidth of the sequence  $\{X_n^i, n \geq 1\}$  ( $i = 1, 2$ ).

The additivity of the effective bandwidth also holds for independent processes of class  $S$ .

This result is very appealing because it suggests that we can treat the various streams as if they had a fixed rate equal to their effective bandwidth. However, we should keep in mind the crude approximations made along the way. We will sharpen these results later.

Summarizing, we have motivated (not really proved) the following result.

**Theorem 9.4.2** Consider a discrete-time queue with  $X_n$  arrivals at time  $n$  ( $n \geq 0$ ) that serves  $c$  customers per unit of time in the queue. If the sequence of arrivals is of class  $S(\Lambda^*)$ , then the invariant probability  $P(W > B)$  that the queue length  $W$  exceeds  $B$  is such that, for large  $B$ ,

$$\frac{1}{B} \log P(W > B) \approx - \min_{a>c} \frac{\Lambda^*(a)}{a - c}.$$

Moreover, the minimum value of  $c$  such that the right-hand side of the above inequality is at most  $-\delta < 0$  is called the *effective bandwidth* of the arrival sequence and is equal to  $\alpha(\delta)$ , where

$$\alpha(\delta) = \frac{\Lambda(\delta)}{\delta}$$

with  $\Lambda(\delta)$  defined by (9.41).

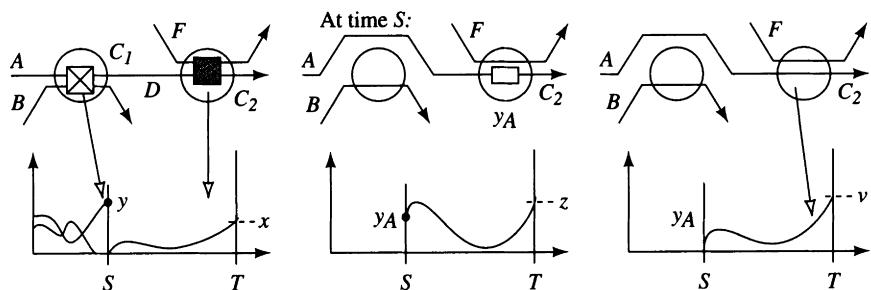
In particular, the effective bandwidth of the sum of two independent arrival sequences is the sum of their effective bandwidths.

Loosely speaking, we can think of the random arrival streams as having constant arrival rates given by their effective bandwidths. This effective bandwidth has a value between the mean and the peak rate of the stream, and it measures in a precise way the burstiness of that stream.

### Network Large Deviations

Consider the network shown in the left-hand part of Figure 9.12. The two queues are first come, first served. The arrival processes  $A$ ,  $B$ , and  $F$  are independent and have straight-line large deviations with effective bandwidth  $\alpha_A(\delta)$ ,  $\alpha_B(\delta)$ , and  $\alpha_F(\delta)$ , respectively. We want, for  $x \gg 1$ , the probability of overflow in the two queues to be dominated by  $\exp\{-\delta x\}$ , where  $x$  is the capacity of each of the two queues. We expect this bound to be valid if the transmission rates  $C_1$  and  $C_2$  of the two queues are large enough.

From the result for the first queue, we know that if  $C_1 > \alpha_A(\delta) + \alpha_B(\delta)$ , then the loss probability in the first queue is bounded by  $\exp\{-\delta x\}$  for  $x \gg 1$ . However, the condition may be more complicated for the second queue.



9.12

Network and modifications.

FIGURE

Indeed, the process  $D$  (see figure) is a complex combination of  $A$  and  $B$  that depends on the service rate  $C_1$ . The large deviations of  $D$  depend on those of  $B$ . For instance, if  $B$  has a large burst of arrivals, then these arrivals create a backlog of arrivals from  $A$  in the first queue. Eventually, when the arrivals of  $B$  are served by the first queue, the backlog produces a burst of  $D$ . Consequently, if  $D$  has an effective bandwidth, it almost certainly depends on the statistics of  $B$ . This discussion shows that we should expect the condition on  $C_2$  to make the losses in queue 2 rare enough to depend in a complicated way on the statistics of  $A$ ,  $B$ , and  $F$ .

However, we can derive a simple sufficient condition, which we state below.

**Theorem 9.4.3** If  $C_1 > \alpha_A(\delta) + \alpha_B(\delta)$  and  $C_2 > \alpha_A(\delta) + \alpha_F(\delta)$ , then the loss probability in each of the two queues is not larger than  $\exp\{-\delta x\}$  for  $x \gg 1$ .

The intuitive meaning of this theorem is that, although the fluctuations of the second queue depend on those of queue 1, the assumptions make the fluctuations of both queues unlikely. The proof of the theorem is a bit involved, but the main idea is rather straightforward and is as follows.

Assume that the arrival processes conspire to make the second queue reach a value  $x$  at time  $T$ . (See left-hand part of the figure.) Denote by  $S$  the last time before  $T$  that queue 2 is empty (with  $S = 0$  if it never is). Say that the processes make the first queue reach some value  $y$  at time  $S$ . The claim is that if  $A$  and  $F$  were sent directly to the second queue during  $[S, T]$ , then they would make it go from 0 to at least  $x - y$ . If this claim is true, then the probability of the behavior during  $[0, S]$  is at most  $\exp\{-\delta y\}$  and that of the behavior during  $[S, T]$  is at most  $\exp\{-\delta(x - y)\}$ . Consequently, the behavior during  $[0, T]$  has a probability at most  $\exp\{-\delta x\}$ .

To fill up the details of the proof, we need to establish the claim and justify the use of independence of the behaviors during  $[0, S]$  and  $[S, T]$ . These behaviors are not independent, but the large deviation behaviors in these two intervals are independent. The independence of the large deviation behaviors is obtained through a continuity argument and the contraction principle, which reduce the calculation of the exponent of the probabilities to action integrals over  $[0, T]$ . The integral being additive, these properties legitimize the use of the product of the probabilities.

The claim itself is proved by the following coupling argument. The first queue contains  $y$  units at time  $S$ . Let  $y_A \leq y$  be the units that arrived from  $A$  among these  $y$  units. At time  $S$ , we place these  $y_A$  units into queue 2 and we attach the arrival process  $A$  directly to queue 2 instead of it having to go through queue 1. (See the center part of the figure.) This modification speeds up the

arrivals of  $D$  into queue 2. Since queue 2 reached the value  $x$  at time  $T$  before the modification without becoming empty during  $[S, T]$ , it must also reach at least the value  $x$ . Indeed, the queue serves  $C_2(T - S)$  during  $[S, T]$  before the modification and cannot serve more after. Thus, the value  $z$  that it reaches is such that  $z \geq x$ . Now, consider what happens if the processes  $A$  and  $F$  arrive directly into an empty second queue at time  $S$ . (See the right-hand part of the figure.) In that case, the number of arrivals in the second queue during  $[S, T]$  is the same as in the case of the center part of the figure. Also, in that case, the queue cannot serve more than  $C_2(T - S)$  during  $[S, T]$ . Consequently, at time  $T$ , the second queue occupancy  $\nu$  will be such that  $\nu \geq z - y$ . Hence,  $\nu \geq x - y$ , as claimed.

This argument extends to feed-forward networks, by induction on the number of queues. We let the reader provide the details.

### *Decoupling Bandwidth*

Consider once again the network of Figure 9.12 but assume that  $F = 0$ . In the above discussion, we argue that if  $C_1$  and  $C_2$  exceed the sum of the effective bandwidths of the arrival processes that go through these queues, then the loss probability is asymptotically small enough. This argument provides a sufficient condition for the second queue. Intuition suggests that the condition may be overly conservative. Indeed, it is possible that the first queue smoothes out the process  $A$  and that  $C_2$  does not need to be as large as  $\alpha_A(\delta)$ .

However, it can be shown that if  $C_1$  is large enough, then  $C_2$  must be at least  $\alpha_A^*(\delta)$  for the loss probability in the second queue to decay as fast as  $\exp\{-\delta x\}$ . The intuition is that if  $C_1$  is large enough, then the first queue does not really smooth out the process  $A$ . Specifically, if

$$C_1 \geq \alpha_A^*(\delta) + \alpha_B(0),$$

then we need  $C_2 \geq \alpha_A(\delta) + \alpha_F(\delta)$  for the loss rate in queue 2 to decay fast enough with  $x$ .

In the above inequality,  $\alpha_A^*(\delta)$  is a value that we call the *decoupling bandwidth* of process  $A$ . We refer the reader to the references for details on these results.

#### 9.4.5 Bahadur-Rao Theorem

We used this theorem in Chapter 8 to estimate the statistical multiplexing gain. We sketch a derivation of the result.

**Theorem 9.4.4** Let  $\{Y_n, n \geq 1\}$  be iid random variables with  $\Lambda(\theta) := \log M(\theta)$ , where  $M(\theta) := E[\exp(\theta Y_1)]$ . Then

$$P(Y_1 + \cdots + Y_N > Nc) \approx \frac{1}{\sqrt{2\pi}\sigma_c\theta_c\sqrt{N}} \exp\{-N\Lambda^*(c)\}. \quad (9.45)$$

In the above expression,  $\theta_c$  achieves the maximum in

$$\Lambda^*(c) = \sup_{\theta} [\theta c - \Lambda(\theta)]$$

where

$$\sigma_c^2 = \Lambda''(\theta_c).$$

**Proof** Define the iid random variables  $Z_n$  so that

$$P(Z_n \in (x, x+dx)) = \frac{\exp\{\theta_c x\} P(Y_n \in (x, x+dx))}{M(\theta_c)}.$$

Note that  $EZ_n = c$  and  $\text{var}(Z_n) = \Lambda''(\theta_c) = \sigma_c^2$ . With  $Z^N = Z_1 + \cdots + Z_N$ ,  $Y^N = Y_1 + \cdots + Y_N$ ,  $\mathbf{Y} = (Y_1, \dots, Y_N)$ , and  $\mathbf{Z} = (Z_1, \dots, Z_N)$ , we find

$$\begin{aligned} P(Y^N > Nc) &= E_{\mathbf{Y}}[1\{Y^N > Nc\}] = E_{\mathbf{Z}}\left[1\{Z^N > Nc\} \frac{P_{\mathbf{Y}}}{P_{\mathbf{Z}}}\right] \\ &= M(\theta_c)^N E[\exp\{-\theta_c Z^N\} 1\{Z^N \geq Nc\}] \\ &= \exp\{-N\Lambda^*(c)\} E[\exp\{-\theta_c W^N\} 1\{W^N \geq 0\}] \end{aligned}$$

where  $W_n = Z_n - c$  and  $W^N = Z^N - Nc$ . Let  $V = W^N / (\sigma_c \sqrt{N})$ . Then

$$P(Y^N > Nc) = \exp\{-N\Lambda^*(c)\} E[\exp\{-\theta_c \sigma_c \sqrt{N} V\} 1\{V \geq 0\}].$$

By the central limit theorem, we know that  $V$  is approximately Gaussian with zero mean and unit variance. We then write

$$\begin{aligned}
P(Y^N > Nc) &\approx \exp\{-N\Lambda^*(c)\} \frac{1}{\sqrt{2\pi}} \int_0^\infty \exp\{-\theta_c \sigma_c \sqrt{N}x\} \exp\{-x^2/2\} dx \\
&= \exp\{-N\Lambda^*(c)\} \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{1}{2}(\theta_c \sigma_c \sqrt{N})^2\right\} \int_0^\infty \exp\left\{-\frac{1}{2}(\theta_c \sigma_c \sqrt{N} + x)^2\right\} dx \\
&= \exp\{-N\Lambda^*(c)\} \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{1}{2}(\theta_c \sigma_c \sqrt{N})^2\right\} \int_{\theta_c \sigma_c \sqrt{N}}^\infty \exp\left\{-\frac{x^2}{2}\right\} dx \\
&\approx \exp\{-N\Lambda^*(c)\} \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{1}{2}(\theta_c \sigma_c \sqrt{N})^2\right\} \frac{1}{\theta_c \sigma_c \sqrt{N}} \exp\left\{-\frac{1}{2}(\theta_c \sigma_c \sqrt{N})^2\right\} \\
&= \frac{1}{\sqrt{2\pi N \sigma_c \theta_c}} \exp\{-N\Lambda^*(c)\},
\end{aligned}$$

as we wanted to prove. In the next to last equality we used the approximation

$$\int_y^\infty \exp\left\{-\frac{x^2}{2}\right\} dx \approx \frac{1}{u} \exp\left\{-\frac{u^2}{2}\right\}, \text{ for } u \gg 1.$$

## 9.5

## SUMMARY

The performance evaluation of circuit-switched networks requires computation of the blocking probabilities; the evaluation of network-congestion and flow-control strategies in packet-switched networks requires computing queue length distributions; and the evaluation of (small) buffer overflow probabilities and multiplexing gain in virtual circuit networks requires computing “tail” probabilities. This chapter provided a rapid but reasonably complete summary of the mathematical models and results underlying all these computations.

The results on blocking probabilities first appeared 70 years ago; those on queue length distributions started in the 1960s; and the first results on tail length distributions were published in the late 1980s. These last results are available only in journals and conference proceedings.

These results are difficult to comprehend fully without a strong background in probability theory. However, it is worth the effort to understand the results, because they provide ways of analyzing resource allocation and admission control for high-performance virtual circuit networks. An indirect method of control is through pricing of network services. That approach is considered in Chapter 10.

**9.6****NOTES**

For general discussions of queuing theory see [K75, K79, W88]. Also see the references cited in section 8.6. More extensive treatments of the deterministic approaches of section 9.4.1 are given in [C91, LV95]. For elaboration of the intuition presented at the end of section 9.4.2, consult [DZ93]. For a discussion of the Bahadur-Rao theorem see Theorem 1.3 of [D91].

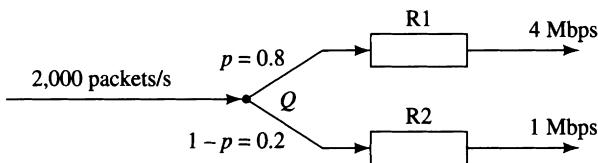
**9.7****PROBLEMS**

1. Let  $\mathbf{x} = \{x_n, n \geq 0\}$  be a discrete-time Markov chain with transition probability matrix  $P$  and invariant distribution  $\pi$ . Show that  $\{z_n = (x_n, x_{n+1}), n \geq 0\}$  is a Markov chain. Find its invariant distribution.
2. Consider a continuous-time Markov chain on  $\{1, 2, 3\}$  with  $q(1) = q(1, 2) = a > 0$ ,  $q(2) = q(2, 3) = b > 0$ ,  $q(3) = q(3, 1) = c > 0$ . Find the rate matrix of the Markov chain reversed in time.
3. Let  $\mathbf{A} = \{A_t, t \geq 0\}$  and  $\mathbf{B} = \{B_t, t \geq 0\}$  be two independent Poisson processes with rate  $\lambda$ . Define  $x_t = x_0 + A_t - B_t$  where  $x_0$  is a random variable independent of  $\mathbf{A}$  and  $\mathbf{B}$ . Show that  $\mathbf{x} = \{x_t, t \geq 0\}$  is a Markov chain. Is it irreducible? Does it have an invariant distribution?
4. Consider a continuous-time Markov chain  $\mathbf{x} = \{x_t, t \geq 0\}$  with rate matrix  $Q = \{q(i, j), i, j \in \mathcal{X}\}$ . Assume that  $\nu$  is the invariant distribution of  $\{\xi_n, n \geq 0\}$ , the Markov chain  $\mathbf{x}$  observed at its jump times. Relate  $\nu$  to  $\pi$ , the invariant distribution of  $\mathbf{x} = \{x_t, t \geq 0\}$ .
5. Consider a simple telephone network where calls that arrive as a Poisson process with rate  $\lambda$  are sent with probability  $p$  to a group of  $M$  lines and with probability  $1 - p$  to another group of  $N$  lines. Find the value of  $p$  that minimizes the blocking probability of a call. Compare the resulting blocking probability to that corresponding to the calls being served by  $M + N$  lines.
6. Consider  $K$  telephone sets that share  $N < K$  outgoing lines of a PBX. Assume that each telephone generates a new call, when it is not busy, as a Poisson process with rate  $\rho$ . Calls have independent holding times that are exponentially distributed with rate 1. Model the number of calls in progress by a Markov chain and calculate the probability that the  $N$  lines are busy. Relate that probability to the probability that a call is blocked. Are these probabilities the same?

7. Show that the blocking probability in problem 6 is insensitive with respect to the distribution of the call holding times.
- Hint:* Consider a closed network with an  $M/GI/\infty$  and an  $M/M/\infty$  queue.
8. Consider  $K$  telephone sets that are independently busy or idle with probabilities  $p$  and  $1 - p$ , respectively. Use the central limit theorem to estimate a number  $N$  such that the probability that more than  $N$  phones are busy is about 1%. Relate this result to that of problem 6.
9. Consider the following model of a fixed wireless network. There are  $N$  radio transmitters that share  $M$  radio channels. Each radio station can serve  $K$  telephone sets. Each telephone set generates calls with rate  $\rho$  when it is not busy. Find a Markov description of the network when the call holding times are independent and exponentially distributed with rate  $\mu$ . What is the blocking probability of this network? Design a Monte Carlo procedure for estimating the call blocking probability.
10. Let  $\{A_t, t \geq 0\}$  be a Poisson process with rate  $\lambda$ . Define the processes  $\{B_t, t \geq 0\}$  and  $\{C_t, t \geq 0\}$  as follows. At each arrival time of  $\{A_t, t \geq 0\}$ , one flips a coin. With probability  $p$ , the outcome is heads and the arrival time is defined to be  $\{B_t, t \geq 0\}$ . Otherwise, that arrival time is defined to be  $\{C_t, t \geq 0\}$ . Show that the processes  $\{B_t, t \geq 0\}$  and  $\{C_t, t \geq 0\}$  are independent Poisson processes with rate  $\lambda p$  and  $\lambda(1 - p)$ , respectively.
11. Assume the times between bus arrivals at a bus stop are equal to 10 minutes with probability 0.5 and to 30 minutes with probability 0.5.
- How long is the average wait for a bus for someone who shows up at a random time at the bus stop?
  - Assume that passengers show up at the bus stop according to a Poisson process with rate 0.4 passengers per minute. What is the average number of passengers in a bus?
12. If the inter-arrival times of buses are equally likely to be 10 and 40 minutes, then the average waiting time of a passenger is
- 25 minutes.
  - 12.5 minutes.
  - 17 minutes.
13. Consider a network of queues with Poisson external arrivals and Markov routing as in Figure 9.7. Assume that some queues are  $M/M/1$  and others are  $M/GI/\infty$ . Find the invariant distribution of the network.

14. Consider a router with output rate 155 Mbps. Because of the TCP protocol, packets tend to arrive in batches. To analyze the average delay per packet, assume that all the packets have the same size equal to 1 KB and that packets arrive in groups of 1, 10, and 20 with equal probabilities. The arrivals of the batches form a Poisson process. The arrival rate at the router (in bits per second) is 80% of the output rate. Calculate the average delay per packet. *Hint:* First treat a batch as a single “big” packet and calculate the average delay per big packet. Next, derive the average delay per regular packet.
15. Consider a router with an output line rate of 1.5Mbps. Two streams of packets arrive as independent Poisson processes. The streams have rates  $L_1 = 500$  packets/s and  $L_2 = 100$  packets/s, respectively. The packets of the first stream have 200 bytes. Packets of the second stream have independent sizes and are 100-byte long with probability 0.5 and 1,000-bytes long with probability 0.5.
  - (a) Calculate the average delay per packet when the router serves the packets in the first come, first served order.
  - (b) Calculate the average delay per packet of type 1 and of type 2 when the router serves the packets of type 1 with high priority and packets of type 2 with low priority.
16. Packets arrive in batches of 100 at a link. The average length per packet is 500 bytes. The link transmission rate is 1 Mbps. The average delay per packet is
  - (a) at least 0.2 s.
  - (b) at least 0.4 s.
  - (c) at least 2 ms.
  - (d) at most 0.4 s.
  - (e) none of the above.
17. In a queue with high and low priorities (non-preemptive), as the arrival rate of low priority packets increases, the average delay of high-priority packets
  - (a) increases.
  - (b) does not change.
  - (c) decreases.
  - (d) may increase or decrease.
18. A stream of packets arrives at point Q as a Poisson process with rate 2,000 packets/s. The packets have 400 bits with probability 0.5 and 4,000 bits with probability 0.5. With probability  $p = 0.8$ , each packet is, independently of

the other packets, sent to node R1; it is sent to node R2 otherwise. The output rate of R1 is 4 Mbps, and that of R2 is 1 Mbps.



Calculate the average delay per packet as a function of  $p$ . Hint: The packets arrive as Poisson processes at R1 and R2. Recall the average delay formula for an M/G/1 queue with Poisson arrival rate  $A$  and independent service times distributed as the random variable  $S$ :

$$\text{Average delay} = \frac{A \times E(S^2)}{2\{1 - A \times E(S)\}} + E(S).$$

19. Consider an  $M/M/1$  queue whose buffer size is fixed and finite. We call this an  $M/M/1/k$  queue. (Note that  $k$  is the total number of customers in the system, which includes the one being served in the server. When it is omitted, such as in the case of  $M/M/1$ , it implies that the system can hold any number of customers.) Suppose the queue has Poisson arrivals with mean rate  $\lambda$ , and exponential service times with mean rate  $\mu$ .
- (a) Identify the states of this system and draw the state-transition-rate diagram.
  - (b) Derive the equations needed to solve for the stationary probabilities of the states. Verify that the stationary probabilities are

$$p_n = \frac{\rho^n(1 - \rho)}{1 - \rho^{k+1}},$$

where  $\rho = \lambda/\mu$ .

- (c) We can see that the buffer may overflow, causing loss of customers in this queue. What is the probability that a customer cannot get into the queue and is lost?
  - (d) If we want our loss probability to be small, say  $10^{-3}$ , how big should our buffer be? Here let's assume that  $\rho = 0.5$ .
20. ATM cells arrive at a transmitter equipped with a buffer according to a Poisson process with rate  $\lambda$  (in cells per cell transmission times). What is the maximum value of  $\lambda$  if the average delay must be less than five cell transmission times?

21. An ATM source produces  $A_n$  cells during the  $n$ th cell transmission time. Assume that  $P(A_n = m/p) = p = 1 - P(A_n = 0)$ . Calculate how many such sources can go through a transmitter equipped with a buffer if the average delay per cell must be less than five cell transmission times.
22. Let  $\mathbf{x} = \{x_t, t \geq 0\}$  be a Markov chain on a finite-state space  $\mathbf{X} = \{1, 2, \dots, m, m+1\}$  with rate matrix  $Q$ . Assume that  $P(x_0 = i) = \pi(i)$ ,  $i \in \mathbf{X}$ . Define  $\tau = \inf\{t > 0 | x_t = m+1\}$ . Calculate  $E(\exp\{-s\tau\})$ .
- Hint:* Let  $\alpha(i, s) = E[\exp\{-s\tau\} | x_0 = i]$ . Argue that if  $x_0 = i$ , then  $\tau$  is equal to the holding time of state  $i$  plus the time to go from the next state to  $m+1$ . Thus,  $\alpha(i, s) = [1/(q(i) + s)] \sum_j \alpha(j, s) \times q(i, j)$  for  $i \neq m+1$ . Derive a set of equations and solve them to obtain  $E(\exp\{-s\tau\})$ . Explain how to use these ideas to construct a Markov chain model of the M/GI/ $\infty$  queue.
23. A simple network consists of two nodes in tandem with respective service rates  $C_1$  and  $C_2$ . A  $(B_1, R_1)$ -traffic (see section 9.4.1) enters node 1 and continues on through node 2. A  $(B_2, R_2)$ -traffic also enters node 2. Find bounds on the delay of stream 1 through the network.
24. Let  $\{X_n, n \geq 1\}$  be a sequence of iid Poisson random variables with mean  $\lambda$ . Calculate the effective bandwidth of that sequence. Compare that effective bandwidth to that of an iid sequence of  $\{0, A\}$  random variables with the same mean rate.
25. Let  $\{X_n, n \geq 1\}$  be iid random variables with values in  $\{0, 1\}$ . Think of these as being on-off sources. Find how many sources can go through a transmitter with rate  $C$  using the Bahadur-Rao approximation. Compare your answer with the result that you would obtain if the sources were Gaussian with the same mean and variance. Assume  $C = 20$ ,  $p = 0.2$ , and  $p_e = 10^{-8}$ .

# Network Economics

I

In this chapter you will be introduced to the basic concepts and models that help to explain some important features of the economy of communication networks. That economy is diversifying and growing rapidly. It includes the markets for communication services, on-line alternatives to traditional markets, markets for information goods that did not exist before, and the market for networking equipment. This chapter is limited to the market for communication services.

Like the study of other economic commodities, the study of the communication services markets focuses on characteristics of supply and demand and on their interaction in the market.

The most important supply factors are the technology of network elements (communication links and switches) and the management rules that can be used to control these elements so that the network supplies the communication services that users want. The technology of network elements is described in Chapters 11 and 12, while Chapters 8 and 9 are devoted to a study of network control. A more complete study of supply should include an assessment of costs: the capital costs of network elements (hardware and software) and operating costs (maintenance, depreciation, administration). We cannot discuss costs quantitatively because they are not published in the open literature.

As the costs of communication and computing hardware decline each year, operating costs become relatively more important. These costs may lock customers to a particular hardware or software manufacturer or service provider, because customers can face high administrative and training costs should they switch to a different supplier or technology. This *lock-in effect* can

often explain the expensive efforts of suppliers of equipment and services to increase their market share as quickly as possible.

The lock-in effect also explains the entrenched nature of TCP/IP. Suppose you have developed a transport protocol, SmartTP, that is clearly better than TCP for Web traffic. To deploy your protocol, however, requires changes in client and server software. The more servers deploy SmartTP, the greater the incentive for clients to do so, and vice versa. This is an example of network externality. The externality creates a positive feedback with a critical size: once the SmartTP installed base exceeds this size, the demand for SmartTP will grow rapidly, but below this size, there is insufficient incentive to deploy SmartTP. As the installed base of the incumbent technology, TCP/IP in this example, grows, the critical size that SmartTP must overcome increases, too.

On the other hand, a software layer on top of TCP/IP that, for example, opens multiple connections or maintains a persistent connection that improves Web access faces a lower critical size, even though the performance gain is much lower than SmartTP.

Demand factors determine the trade-off users make between the services they want, their quality, and their price. The demand for communication services in large part is *derived demand*. That is, users demand communication services not for themselves, but because those services provide a means to an end that users really want. For example, users may access an on-line database in order to obtain news; or they may subscribe to a video-on-demand service to watch a movie; or they may telecommute. In each case, users have alternative means to achieve their ends: news may be obtained from a newspaper, radio, or TV; videos may be rented from a store; and one may travel to work. The price and convenience of use of these alternatives will affect the derived demand.

We begin in section 10.1 with a discussion of the nature of derived demand. In section 10.2 we analyze the market for Internet access. We argue that flat rate pricing is inefficient and retards the introduction of quality-differentiated services.

In section 10.3 we introduce a resource model of network access to suggest four types of charges for communication services. Implementing these charges requires a technology capable of service provisioning, accounting and billing, and user control that does not yet exist. In section 10.4 we present results from a trial at the University of California, Berkeley, called INDEX (Internet Demand Experiment). INDEX is both a technology and a market trial. It demonstrates a technology with the capabilities mentioned above and collects data about how users value different qualities of service.

In sections 10.5 and 10.6 we introduce two models to study pricing of communication services. The first model is useful for a data network like the

Internet, with a single quality of service and no admission control, making the network susceptible to congestion. The second model is for networks where service quality guarantees are possible and new user requests can be rejected if network resources to serve those requests are not available. We use the second model to calculate how many requests of different types of service can simultaneously be served. We also develop a variant of this model to suggest that a good way to charge for those services is in terms of the resources they consume.

Section 10.7 provides a summary.

## **10.1 DERIVED DEMAND FOR NETWORK SERVICES**

Communication networks, like road networks, alter the absolute and relative cost of access to locations of activities that people value. Unlike physical locations that are reached by a road network, the locations reached by a communication network are hosts connected to the network. These hosts are *virtual locations*.

Physical and virtual locations provide the "space" for a variety of activities. Some physical locations are workplaces, others are shops or places that provide entertainment, and others are homes where people live. Virtual locations (Web sites, workplace hosts, e-mail servers, etc.) provide information or the means to buy and sell goods and services on-line.

When people embark on an activity at a remote physical or virtual location, they incur a communication cost and a search cost. These costs are subtracted from the value of the activity itself. If you purchase a book from a physical bookstore, the communication cost is the cost of transportation to one or more bookstores, and the search cost is the time and effort you spend in finding the book you want. If you go to work or to visit a friend, the communication cost is the effort in traveling to your destination, and the search cost is negligible. If you access an on-line library catalog to research some topic, the communication cost may be the price you pay for the communication services, and the search cost is determined by how well the on-line search facility matches your need.

The sum of the communication and search costs is the *transaction cost* associated with an activity such as going to work, shopping for a book, or Web browsing for some information. The transaction cost has a monetary component, and a subjective component, which is a function of how the transaction is

experienced. If the transaction uses network services, the experience depends on the time it takes to complete the transaction such as downloading a Web page. That time is determined by the speed of the service. Other dimensions of service quality (latency, jitter) also affect the experience. Economists suppose that an individual assigns a monetary value to this subjective element of the transaction cost. When we refer to the transaction cost faced by an individual, we mean the sum of the monetary cost and this assigned value.

The subjective cost of the same transaction is different for different individuals and for the same individual at different times. For example, when individuals place a high value on their time, they will choose a more expensive, high-speed service because of the time they save.

Individuals may have a choice among physical and virtual locations, and among communication services of different qualities and price. They compare the total cost of the different alternatives and choose the one with the least cost.

Consider the choice between physical and virtual locations. For some activities the transaction cost at a virtual location is much lower than at a physical location. So people are increasingly likely to visit virtual locations for those activities. This explains the growth of on-line commerce, especially for goods and services that are reasonably standard (purchasing airline tickets, books, or stocks).

The shift to on-line commerce is altering many businesses. Airline travel agents have seen their revenues decline precipitously. Soon there will be few of them left as travelers save money and time through on-line purchases. The growth of on-line trading in stocks because of its convenience and much lower transaction cost is reducing the incomes of brokers. On-line "aggregators" are attracting customers by providing information about price and quality of products from competing manufacturers. In 1998 1% of total retail trade in the United States was conducted on-line, and some estimate that this will increase to 6% by 2002. On-line sales among businesses will increase even faster judging by the examples of Cisco, Dell Computer, and Boeing.

### 10.1.1 Information Goods

The growth in on-line commerce is attributable to the reduced transaction cost. Total cost reductions are even larger for on-line sales of *information goods*. These goods are themselves in digital form (e.g., software packages and database access). This cost reduction will stimulate the production of many goods such as CDs, videos, and books in digital form.

We point to some distinguishing features of the market for information goods. The costs of producing, distributing, and consuming information goods differ from those of other goods and services. While it can be costly to design and produce the first (or master) item, the cost to produce additional items or distribute them on-line is negligible.

The consumer of an information good incurs, in addition to the cost of the good itself, the training and administrative cost of "absorbing" the good. For example, the cost of using a new software package—one that is different from the one you are currently using—can be high and serve as a deterrent from acquiring the new package. In some cases, the cost of switching to a new package is high because it requires others to switch as well, as is the case with document packages.

The high initial cost, the low marginal cost of production and distribution, and the high absorption cost affect the nature of the competition in the markets for information goods and the strategies that producers and consumers of those goods are likely to follow. We note some of these impacts.

Because the marginal cost of production is low, price competition can lead to pricing below the average cost. This benefits consumers but it can ruin producers. In this situation, a producer may engage in price competition to build up market share. Once an adequate share has been acquired, prices of the same or complementary products could be raised without losing customers who would be reluctant to switch to a competing product because of its absorption cost. The incentive to increase market share explains why Web browsers are given away free. The aim is to sell complementary products such as servers.

Lastly, since the marginal cost of production is low, but consumers face high absorption costs, producers will try price discrimination. They would like to charge a high price to current customers and a low price to attract customers of competitive products. This price discrimination can take a variety of forms. One form is product differentiation. For example, the *New York Times* gives free on-line access to current articles, but you have to pay for archived material. Often an introductory version of a software product is given away free, but advanced versions are sold at a high price.

## 10.1.2

### Site Rents

The road network and the communication network alter the distribution of transaction costs over the geography of physical or virtual locations. Suppose location A is reached at a lower cost than B. Then more people will visit A than B, and so merchants at A will generate higher sales than at B. As a result, rents at A will be higher than at B. The difference in the rents is called *location* or

*site rent.* Landlords of commercial space try to increase site rents by attracting more people to their location.

Site rents also accrue to owners of virtual locations such as Web sites. A common form of site rent is a charge for advertising at a Web site. This is also how TV, radio, and newspapers collect site rents. To attract visitors Web site owners subsidize various services (e-mail, bulletin boards, chat groups), just as TV broadcasts are free and newspapers are sold below cost. Because they compete for the same viewers from the general public, Web sites imitate one another. So there is a tendency for Web sites to look alike, just as TV programs, newspapers, and politicians during elections look alike.

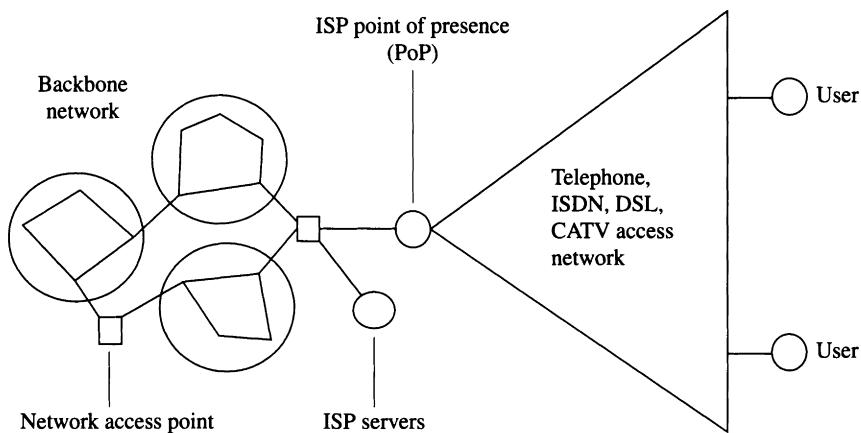
An alternative strategy to increase site rents is to specialize and target a narrow set of viewers. This occurs in physical locations as well, so restaurants tend to cluster together, as do warehouses and investment banks. Similarly, some Web sites specialize in attracting viewers seeking financial information, others attract chemists, and others specialize in information about automobiles or real estate.

There also are some interesting differences between the site rents at physical and virtual locations. For example, since it is possible to determine if an item was purchased from a particular Web site, the site rent may take the form of a commission on the sales. This is not generally possible with newspaper or TV advertising (although it is the case with on-line TV shopping channels). Some Web sites obtain demographic information about their viewers (usually in exchange for a free service like e-mail). They can then target advertising more selectively and charge a higher rent.

The attraction of one physical location relative to another may be because the former is more easily accessible by road. The attraction then persists for a long time as the road network changes very slowly. Virtual locations, by contrast, are likely to be equally accessible and it is easy to build competing locations, so their relative attraction will be determined by how much they reduce the search cost component of transaction costs, how much they subsidize services that viewers desire, and how much brand loyalty they can create.

## **10.2 INTERNET SERVICE PROVIDERS**

In this section we study the market for Internet access. We first investigate a theoretical model and then present some empirical evidence that supports the model.



10.1

FIGURE

An ISP aggregates subscriber traffic and forwards their datagrams to the backbone network through a network access point or NAP.

In the United States, individuals and businesses purchase communication services from an Internet service provider (ISP), as depicted in Figure 10.1. Users connect via a dedicated or shared link to the ISP's point of presence (PoP). The ISP aggregates the traffic at the PoP, where its routers forward user datagrams to the Internet backbone network through a network access provider (NAP). The ISP's subscribers may also connect to ISP servers that provide e-mail, news, and Web-caching.

Routing of packets among backbone NAP networks depends on reciprocal peering arrangements. NAPs do not bill one another for this traffic. But a NAP may use policy-based routing to provide better treatment of traffic from certain peers depending on the reciprocal arrangement. This is different from the telephone network in which the charge for a call is divided among the telephone companies that carry this call according to a system of settlement charges.

There are more than 3,000 ISPs in the United States. Most of them are local with one or a few PoPs. Some have PoPs that span a state or region. A few are national. In some countries the national telephone company is the sole ISP, although this is being replaced by a system of competing ISPs.

A subscriber pays two charges to access the Internet. The first is a charge for the link to the ISP's PoP. Ninety percent of subscribers pay a fixed monthly charge to their telephone company for this link. (In most countries, but not in the U.S., there is an additional telephone connect-time charge that has a large

impact on demand.) In more than 90% of cases this link is an analog voice line used with a dial-up modem at speeds of 28 or 56 Kbps. A growing number of subscribers have a dedicated 128 Kbps ISDN line or a DSL line that provides access at downstream/upstream speeds of 384/128 or 1,500/384 Kbps. Large organizations, with thousands of host computers, may lease lines at speeds of 1.5 to 45 Mbps. A recent alternative is offered by CATV operators. In this case up to 500 subscribers share a 10/3.8 Mbps link (this speed will increase in the future). There are other alternatives as well, including wireless and satellite links. The fixed monthly charge for the link is \$15 to \$20 for an analog voice line, \$40 to \$200 for an ISDN or DSL line depending on speed, and \$30 for CATV (in addition to the \$30 monthly charge for the TV programs). The subscriber pays for the modem. (In 1999, about 600,000 subscribers in the U.S. had CATV and DSL access.)

The subscriber also pays a charge to the ISP that depends on the access speed. Most ISPs in the United States have moved from a connection time-based charge to a flat-rate monthly charge. Depending on the access speed, this charge ranges from \$20 for 28 Kbps and CATV to \$200 for 1.5 Mbps.

The ISP incurs two costs. The first is the cost of aggregation. If it provides access via dial-up modems, the ISP must purchase a pool of modems to terminate subscriber calls and pay the telephone company for trunks to the modem pool. The amortized cost for one modem and trunk line is about \$25 per month. With a concentration of 10 subscribers per modem, this amounts to a cost to the ISP of \$2.50 per month per subscriber. In the case of CATV access, user traffic is already aggregated. DSL traffic is aggregated at the DSL multiplexer where the DSL loops terminate.

The ISP must also pay the NAP for transport of its traffic over the backbone. NAPs offer sophisticated service plans ranging from a fixed monthly charge depending on speed to a charge that depends on the average traffic rate plus a burst charge. They also offer other services such as virtual private networks, reliability, tunneling, and security. The NAP charge typically is less than 1 cent per MB of backbone traffic.

The subscriber's cost of data transfer is high. In the U.S. the average amount of data transferred via 28 Kbps modems is about 60 MB per month, so at a monthly cost of \$20, the subscriber is paying 33 cents/MB. If you include the monthly \$15 for a telephone line, this becomes 58 cents/MB. The light user, who is transferring one-fifth of the average traffic, is paying an exorbitant \$1.65 to \$2.91 per MB, and the heavy user, who transfers five times the average, pays 6.6 to 11.6 cents/MB.

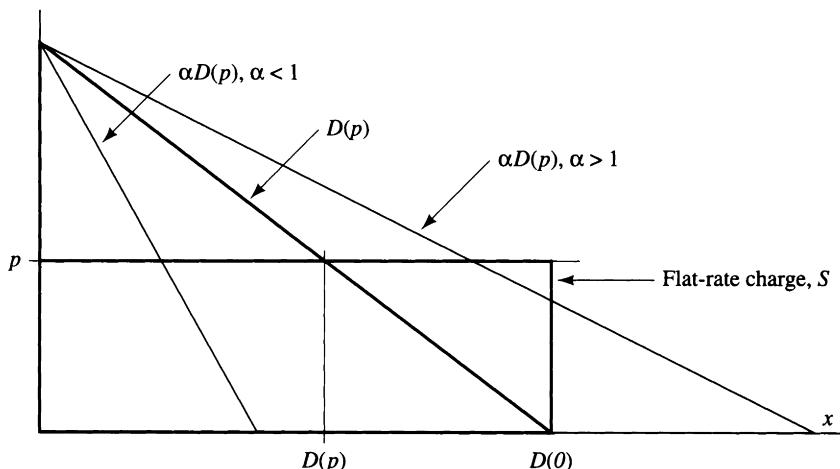
Data from one large ISP indicates that 76% of the traffic uses the HTTP protocol; 12% uses NNTP (Network News Transfer Protocol, a protocol for the

distribution, retrieval, and posting of news articles); 5% is e-mail (POP3 and SMTP); 3% uses FTP; and 2% uses a secure HTTPS (secure HTTP).

### 10.2.1 A Subscriber Demand Model

We develop a model that we will use to compare the behavior of ISPs and their customers under two pricing schemes: a monthly flat-rate charge and a usage-based charge.

Suppose an ISP sells Internet access at a price of  $p$  per unit of usage, measured in minutes of connect time or in MB of data transferred. The average user's demand for access is modeled as a function  $x = D(p)$ . This is the decreasing *demand curve* in Figure 10.2. The user, faced with a unit price  $p$ , does not consume more than  $D(p)$  because the value or benefit she gains from any additional usage is worth less to her than her cost  $p$ . She does not consume less, because her benefit from each unit of usage up to  $D(p)$  exceeds  $p$ . That is the meaning of a demand curve. (If the usage is measured in connect time, and the subscriber must also pay a per-minute connect-time charge  $\tau$  to the telephone company, the demand curve is  $D(\tau + p)$ .)



At a unit price  $p$ , the average user consumes  $D(p)$  units; a light user consumes  $\alpha D(p)$  with  $\alpha < 1$ ; a heavy user consumes  $\alpha D(p)$  with  $\alpha > 1$ .

If access is sold at  $p$  per unit, the user purchases  $D(p)$ . The value to her of consuming  $D(p)$  units, minus her cost  $pD(p)$ , equals the area of the triangle below the demand curve and above the horizontal line  $p$ . This area is called the *consumer surplus*. (If the ISP's unit cost of providing access is  $p$ , then the consumer's cost  $pD(p)$  equals the ISP's cost, and the consumer surplus equals the gain in social welfare.)

Different subscribers have different demand curves  $D_i$ . For simplicity, we suppose that these demands differ by a scale factor, so  $i$ 's demand is  $D_i(p) = \alpha_i D(p)$ . For a heavy user,  $\alpha_i > 1$ , and for a light user,  $\alpha_i < 1$ .

### **Flat-Rate Charge**

Suppose the ISP switches to a flat-rate charge of  $S$  per month. This flat-rate scheme is likely to induce four changes.

First, after paying  $S$  the subscriber faces no additional charge, so the incremental price is 0. As a result,  $i$ 's consumption will increase to  $\alpha_i D(0)$ . This is wasteful since each of the  $\alpha_i [D(0) - D(p)]$  additional units of consumption is worth less to her than the cost  $p$  of producing it. Note that heavy users (with  $\alpha_i > 1$ ) waste more than light users. Also note that the more price-sensitive the user is, the larger  $D(p) - D(0)$  will be, and the greater the waste.

To explain the second effect, suppose that the ISP sets the fixed charge  $S$  to cover the cost. Since the average demand curve under flat-rate charge is  $D(0)$  and  $p$  is the unit cost,

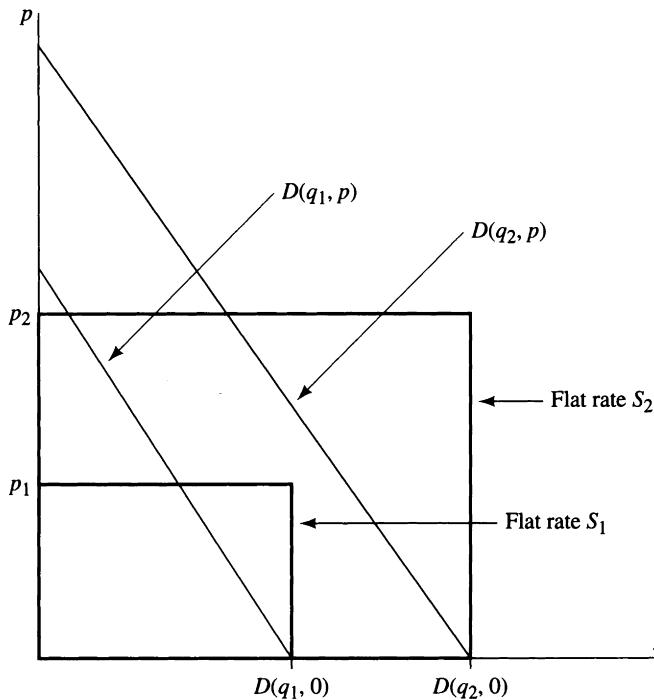
$$S = pD(0).$$

This is the rectangular area indicated in Figure 10.2. So  $i$ 's bill now is  $S$  compared with her previous bill of  $p\alpha_i D(p)$ . If

$$\alpha_i < \frac{D(0)}{D(p)},$$

customer  $i$  will end up paying more. Since  $D(0)/D(p) > 1$ , the average customer will pay more under a flat rate. If a small fraction of the customers are heavy users (with large  $\alpha_i$ ), then most customers will end up paying more and subsidizing high-usage customers.

The increased cost to low-usage customers will induce them to leave, while the subsidy for high-usage customers will attract more such customers. To counter this tendency, the ISP will try to retain its customers by lowering its fixed charge to below cost or by offering other services. The lower the fixed charge, the larger the number of customers, and the greater the ISP's loss from



10.3

FIGURE

Demand increases with quality,  $q$ . ISPs will offer flat-rate tiered service. Light users will not subscribe to higher-quality tiers.

providing Internet access. To cover the loss, the ISP will try to use its customer base to increase site rents and other revenues. Thus the third impact is that ISPs will reorient their business. Since site rents are greater the more time subscribers spend at the ISP's Web sites, the ISP will tend to make it more difficult to explore virtual locations other than the ISP's own Web sites.

The fourth impact is on quality. To explain this, we suppose that the demand is also a function of quality,  $q$ , of access. We model  $i$ 's demand by the function  $x_i = \alpha_i D(q, p)$ , as in Figure 10.3. To fix ideas, we think of quality as speed—the higher the speed, the more users will access the Internet.

The quality of service depends on the ISP's physical capacity  $C$ , the number of subscribers  $n$ , and the average demand  $x$ . This is modeled as

$$q = Q(C, n, x).$$

$Q$  is increasing in  $C$  and decreasing in  $n$  and  $x$ . Under usage-based pricing, the quality level  $q_u$  is given by

$$q_u = Q(C, n, D(q_u, p)).$$

If the ISP does not increase  $C$  and keeps its customer base after switching to flat rate, the quality level will change to  $q_f$ ,

$$q_f = Q(C, n, D(q_f, 0)).$$

Since  $D(q, p)$  is increasing in  $q$ , these relations imply that

$$q_f < q_u.$$

So a flat-rate charge will lead to quality degradation, unless the ISP makes additional investment in capacity. As the ISPs compete for customers and charge a flat rate, the charge will tend to equalize. This will prevent them from attracting customers by offering better quality. That is, if they invest in capacity and improve the quality compared with their competitors, they will attract their competitor's subscribers to the point where the increased numbers of customers will again lower the quality to its previous level. So flat-rate charges prevent ISPs from offering better quality service even if subscribers are willing to pay for it.

An alternative strategy the ISP can follow is to offer flat-rate tiered service. In terms of Figure 10.3 this means that the ISP offers two (or more) quality levels,  $q_1$  and  $q_2$ , at different flat-rate charges  $S_1$  and  $S_2$ , and users may subscribe only to one quality. If user  $i$  chooses quality  $q_1$  she will consume  $\alpha_i D(q_1, 0)$ , if she chooses  $q_2$  she will consume  $\alpha_i D(q_2, 0)$ . If the ISP's unit cost of providing quality  $q_i$  is  $p_i$ , the flat rates are then given by the rectangular areas  $S_i = p_i D(q_i, 0)$ , as indicated in the figure.

If  $i$  chooses tier  $q_1$ , her benefit will be  $\alpha_i$  times the area of the small triangle, and her cost will be the area of the small rectangle,  $S_1$ . So her surplus will be the difference between those two areas. If she chooses tier  $q_2$ , her surplus will be the difference between  $\alpha_i$  times the area of the large triangle and  $S_2$ . We can see that only the heaviest subscribers will choose the higher-quality tier. On the other hand, with usage-based pricing, a subscriber will consume both qualities of service. Thus the introduction of quality tiers will reduce the number of subscribers that will use the higher speed. This, in turn, will limit the deployment of broadband access and the market for communication services that need higher speed.

How large these four impacts are in practice is an empirical matter. The more diverse the demand among customers (the greater the variation in the  $\alpha_i$ ), the larger will be the cross-subsidy of heavy users. The greater the price sensitivity, the larger will be the resulting waste,  $D(q, p) - D(q, 0)$ , and drop in quality. If both effects are large, the ISP will incur a large loss to maintain its customer base, and the more it will rely on site rents for revenues.

### 10.2.2 Empirical Evidence

Empirical evidence suggests that the impacts discussed above are indeed large. We begin with the experience of two large ISPs. We then look at some more detailed data.

In December 1996, ISP A changed from usage-based pricing (\$9.95 per month, including 5 hours of connect time, plus \$2.95 for each additional hour) to a \$19.95 per month flat rate (later increased to \$21.95). As a result, average monthly connect time per subscriber jumped from 6.4 hours to 22.1 hours in 1998, revenue per hour of connect time declined, as did A's operating margin.

ISP B offers Internet access over CATV at a flat rate. The relatively high bandwidth of the channel (10 Mbps downstream and 1.5 to 3.0 Mbps upstream) is shared by up to 500 customers. This sharing makes the service particularly vulnerable to heavy users. Not surprisingly, the company announced that it had difficulty maintaining the advertised high speed. An industry newsletter reported that 1% of B's customers accounted for 90% of the total traffic. To reduce the deterioration in quality by heavy users, the company limits streaming video connections to 10 minutes. A and B expect to make up their losses from other revenue sources.

These two examples suggest that the ISP flat-rate pricing model might be based on an incorrect analysis. On the one hand, the flat rates charged appear to be low because at those rates the ISPs are operating at a loss. At the same time, competition maintains flat rates at these low levels. Thus, a flat-rate charge can only be sustained if there are adequate collateral sources of revenue (advertising, retail trade). Those sources depend on having large numbers of customers, and this places a further downward pressure on the flat-rate charges. On the other hand, if usage-based charges were introduced, then the analysis above suggests that most customers would face *lower* bills than even the apparently low flat-rate charges, heavy users would reduce their demand, and ISPs would be able to recover their cost.

As anticipated in the theoretical model, ISPs are now offering access at different speeds with tiered, flat-rate charges. The large flat rate significantly

Subdomain	Number of hosts	Per host average			In category (%)	
		Connections	Datagrams	Bytes	Datagrams	Bytes
cs.berkeley.edu	275	315	42,248	8,792,253	22.4	31.5
eecs.berkeley.edu	365	108	14,619	2,347,930	10.3	11.2
cc.berkeley.edu	102	1,743	107,153	7,999,763	21.1	10.6
lib.berkeley.edu	229	208	15,672	1,952,972	6.9	5.8
hip.berkeley.edu	299	41	6,350	951,868	3.7	3.7
astro.berkeley.edu	69	78	18,460	3,614,468	2.5	3.2
icsi.berkeley.edu	64	87	14,518	3,608,875	1.8	3.0
biochem.berkeley.edu	56	161	15,183	3,276,475	1.6	2.4
math.berkeley.edu	63	121	15,499	2,825,546	1.9	2.3
ocf.berkeley.edu	22	397	108,559	7,378,574	4.6	2.1

**10.1**  
**TABLE**

Average usage by hosts in the top 10 subdomains. Table excludes hosts that serve the entire campus. These hosts are mostly in cc (central campus) and hip (home IP) subdomains.

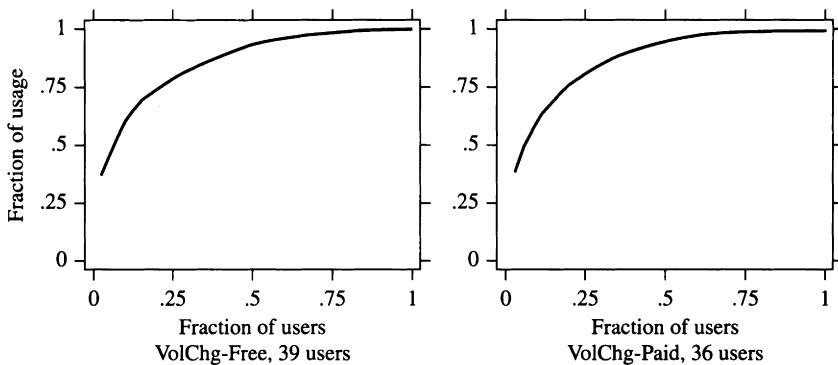
reduces the number of subscribers for the higher-speed tiers. With quality-differentiated, usage-based pricing, however, many more users would demand higher quality service, encouraging the development of applications that require higher-speed access.

### *Diversity in Demand*

That different users have different demand is obvious. More remarkable is a fairly stable 70/20 rule found in several empirical studies: the heaviest 20% of users account for about 70% of traffic.

We consider two empirical studies. In September 1994, a four-day trace was collected of all network traffic on the Berkeley campus backbone network. Approximately 22,000 hosts were then attached to the Berkeley campus, of which 3,172 hosts participated in WAN TCP connections. Table 10.1 shows the usage per host averaged over the trace duration for the top 10 subdomains within the campus. These statistics show significant variation in usage by subdomain. There is an even greater variation by host within each subdomain. For example, the EECS department subdomain with 365 hosts showed an average TCP usage of 2.3 MB and a standard deviation of 17 MB! Indeed, 25% of the heaviest users accounted for more than 80% of the traffic.

INDEX is a market trial that began in April 1998. Customers were offered a range of quality choices at different prices. The left side of Figure 10.4 shows cumulative upstream plus downstream volume of traffic during a week when



10.4

**FIGURE**

Cumulative distribution of traffic volume generated during one week of free service (left) and one week of volume-priced service (right).

service was free. The curve on the right is for a week when customers were charged by volume. Once again 25% of the heaviest users accounted for 75% of the traffic. The data also shows that the high-usage customers are the same whether service is free (as in fixed-rate charge) or when it is priced according to usage. Of course, under usage-based pricing, the volume of traffic declines: during the priced week, customers on average faced a charge of 7 cents per MB and generated 40 MB of traffic, compared with 60 MB of traffic during the free week.

Thus, under usage-based pricing, about 70% of customers would pay less than they would under a fixed-rate charge, and the traffic would decline significantly, as undervalued use is eliminated.

### *Price Sensitivity*

In one INDEX experiment, customers can instantaneously select their access speed (16, 32, 64, 96, or 128 Kbps) and are charged per minute of connect time. (See Figure 10.6 in section 10.4 for the menu offered to customers.) So the demand  $x_k$  of minutes of connect time for the  $k$ th speed is a function of the prices of all speeds,

$$x_k = D_k(p_{128}, p_{96}, \dots, p_{16}).$$

For an empirical study, a particular functional form has to be selected. The following demand model is estimated:

$$\log x_k = b_k + \sum_j \alpha_k^j \log p_j, \quad k, j \in \{16, 32, 64, 96, 128\}, \quad (10.1)$$

where  $x_k$  is the number of minutes of connect time of speed  $k$  Kbps purchased by a consumer during a week when facing a price of  $p_j$  cents per minute for speed  $j$ . With this "log-log" model, the coefficient  $\alpha_k^k$  is the own-price elasticity, that is,  $\alpha_k^k$  is the percent change in demand  $x_k$  for speed  $k$  due to a 1% change in its price  $p_k$ ; and  $\alpha_k^j$  is the cross-price elasticity, that is, the percentage change in the demand  $x_k$  due to a 1% change in the price  $p_j$  of speed  $j$ . The prior expectation is that  $\alpha_k^k < 0$ , demand for speed  $k$  will drop if its price increases, and  $\alpha_k^j > 0$ , demand for speed  $k$  will increase if the price of a substitute speed increases.

The estimated demand equation is

$$\begin{aligned} X_{128} &= 4.1 \quad -\mathbf{1.65}P_{128} \quad +0.44P_{96} \quad +\mathbf{0.55}P_{64} \quad -0.12P_{32} \quad +0.00P_{16} \\ X_{96} &= 2.6 \quad +\mathbf{1.23}P_{128} \quad -3.34P_{96} \quad +1.17P_{64} \quad +0.23P_{32} \quad +0.00P_{16} \\ X_{64} &= 2.7 \quad +0.08P_{128} \quad +\mathbf{0.84}P_{96} \quad -\mathbf{1.71}P_{64} \quad +0.47P_{32} \quad +0.55P_{16} \quad (10.2) \\ X_{32} &= 2.33 \quad +0.48P_{128} \quad -0.58P_{96} \quad +\mathbf{0.88}P_{64} \quad -\mathbf{1.10}P_{32} \quad +0.08P_{16} \\ X_{16} &= 0.52 \quad +0.42P_{128} \quad -0.26P_{96} \quad +0.18P_{64} \quad +\mathbf{0.97}P_{32} \quad -\mathbf{1.29}P_{16} \end{aligned}$$

where  $X_k = \log x_k$ ,  $P_i = \log P_i$ . The estimated coefficients in (10.2) whose  $t$ -static (that is, estimate divided by its standard error) is larger than 3 are written in bold. Observe that the demand for a particular speed is very sensitive to its own price and to the price of the next lower speed. The own-price elasticity is between  $-1$  and  $-3$ , and the cross-price elasticity is around 1. The high own-price elasticity implies that when usage is connect time, waste induced by flat-rate pricing is large.

Second, the large cross-price elasticity indicates that if users are offered differentiated quality of service, they would indeed demand more than one service quality. Direct evidence of this in INDEX data also comes from the fact that on average, each customer selects 3.5 out of the 5 available priced speeds (in addition to the free 8 Kbps speed) each week. Since consumers do value multiple service qualities, there is a loss to providers and consumers when ISPs only offer tiered quality service as they do today. (See section 10.4.2 for more on this topic.)

### **10.3**      NETWORK CHARGES: THEORY AND PRACTICE

In this section we introduce the economic principles underlying charges for communication services. Our examples will largely be drawn from the Internet, although the discussion applies to other communication networks.

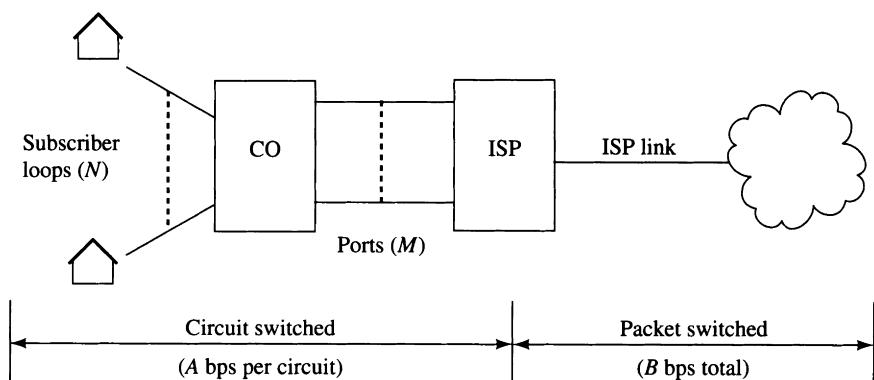
A word on terminology: we will use the terms *charge* and *price* interchangeably, although a useful distinction can be made. A price is normally associated with one unit of service: if you buy  $n$  units of service, you pay  $n$  times the unit price. A charge is a more general form of price. For example, a charge may consist of a fixed component plus a unit price. Mathematically, we may think of a charge as a nonlinear price and a price as a linear charge.

### 10.3.1 A Resource Model

Figure 10.5 is a model of the resources that an ISP needs to provide Internet access to the home. There are four kinds of resources.

Subscribers connect their modems via circuit-switched connections to the ISP modems. This resource is characterized by the bit rate  $A$  bps of the subscriber modem and the number of access ports that the ISP provides. The figure shows that  $N$  end hosts share  $M$  access ports. The third resource is the packet-switched link with bit rate  $B$  bps connecting the ISP to the backbone network via a NAP. The fourth resource is the backbone network itself depicted by the cloud.

These resources and the way they are managed determine the quality of service that a subscriber receives. The bit rate  $A$  on a subscriber's dedicated link limits the peak rate. Dial-up modems over analog loops have speeds of 28 or 56 Kbps. Digital subscriber loops (DSL) permit speeds up to 1.5 Mbps. In some metropolitan areas in the U.S., Internet access accounts for 50% or more of the traffic over the local telephone network. Since  $N$  subscribers contend for  $M$  access ports, a subscriber may be blocked. The blocking probability depends,



**FIGURE**  
10.5

The service quality offered by an ISP depends on the speed of the link, the blocking probability of the modem pool, the congestion in the shared packet link, and the network state.

as we saw in Chapter 9, on  $N$ ,  $M$  and on the holding times of a subscriber's connection. (ISPs typically adopt a concentration ratio ( $N/M$ ) of 10 or more. Holding times depend on whether users face a flat-rate or a connect-time charge. Under flat-rate charges, the average holding time is 50 minutes.)

In the case of CATV, subscribers access the ISP via a shared link, as we saw in Chapter 5. The link provides speeds up to 3 Mbps upstream and 10 Mbps downstream. Since the link is shared by up to 500 subscribers, it may be congested.

The subscribers share the packet-switched link with a peak rate  $B$ , and so the quality depends on how congested this link is. Lastly, the service quality depends on the state of the rest of the network depicted in the figure by a cloud.

In summary, there are dedicated resources such as a subscriber loop, circuit-switched shared resources that may be subject to blocking, packet-switched shared links that may be congested, and the backbone network itself that may be congested.

### 10.3.2 Economic Principles

Subscribers pay a charge for Internet access. Ideally, the charge covers the resource cost. When a resource is dedicated to a particular subscriber, it is clear that its charge should equal its cost. When resources are shared, the charge also serves to ration access so as to limit congestion. It is difficult to assess the proper congestion charge.

Although one may imagine many kinds of charges, it is important to distinguish only four types: an access fee, a usage charge, a congestion charge, and a service quality charge.

An *access fee* is a monthly subscription fee for being connected to the network. The access may be limited to a certain period of time each day, or it may be unlimited. The fee is paid independently of how many connections the subscriber actually makes, or how much data is transferred, that is, it is independent of the subscriber's use of the network. Ideally, the access fee should equal the cost of connecting a user to the network. Thus the monthly fee charged by the telephone company should pay for the local loop connecting the subscriber to the central office. Similarly, for each subscriber an ISP incurs a fixed cost for hardware (modem, host memory, etc.), software, and administration (setting up and managing the customer account) that should be recovered through the access fee.

On the demand side, a user would subscribe to the network only if his willingness to pay exceeds the benefit of access. Because access confers the right to use the network whether or not one actually uses it, users may subscribe to preserve their options. (Common examples of access fees are the

membership dues one pays to join a club or a museum; membership gives one the right to use the club or visit the museum.)

In terms of the resource model of Figure 10.5, the access fee should cover the fixed cost of the subscriber loop, the ISP's facilities, and the backbone network.

A *usage* charge depends on the amount of use. It is, therefore, based on each connection or call the subscriber makes, or the amount of traffic she generates. (We use the terms *connection* and *call* interchangeably to refer to a TCP connection, an ATM virtual circuit connection, or a telephone call.) The usage charge can be calculated in many ways. It may be a function of the duration of the connection, the amount of data transferred, or the end-to-end distance. Telephone usage charges, for example, depend on the call duration and the distance between the calling parties. Usage charges should equate cost and benefit. More network resources (transmission links, buffers, routers or switches, system operations and maintenance, etc.) are devoted to subscribers or connections with greater usage, and so they should pay a greater share of the network cost. Ideally, the usage charge should equal the cost of the additional resources that a connection or call needs.

In the case of Internet access, the usage charge for a connection could include a component that is a function of the bandwidth and proportional to the connect time, and a component that is proportional to the volume of traffic.

A *congestion* charge depends on the amount of traffic or load that the network is carrying at the time of the subscriber's connection. Congestion charges are responsive to the state of the network: the charge is higher when the network is congested, and there is no congestion charge when it is uncongested. The rationale for a congestion charge is as follows. Service quality deteriorates rapidly as the network approaches congestion. In any network that uses statistical multiplexing, such as the Internet or an ATM network, congestion increases queuing delays or packet loss due to buffer overflow. In the telephone network, which uses time-division multiplexing to reserve a fixed bandwidth for each call, congestion increases the blocking probability.

In order to prevent congestion and the resulting quality degradation, the number of connections must be limited. There are many ways to do this, but it would be socially advisable to permit connections network users deem more valuable and prevent connections users think are less valuable. One way to do this is by imposing a congestion charge: users would initiate only those connections that they value more than the charge and postpone making less-valued connections to a time when the congestion charge is low or zero. In this way congestion charges can be used to discourage less-valued connections. The level of congestion charge should be just sufficient to prevent congestion: too

low a charge will not prevent congestion, and too high a charge will reduce the number of calls to a level below what the network can accommodate.

Sometimes the traffic pattern varies regularly over the day, and congestion predictably occurs during certain busy hours of the day. In these cases a congestion charge may be approximated by a "time-of-use" charge: there is a higher usage charge during busy periods. (The difference between the usage charge during the busy and nonbusy periods is a congestion charge.) This is what telephone companies do when they impose a lower usage charge at night and on weekends.

In the Internet, congestion has two different time scales. There is a predictable pattern of congestion that lends itself to a time-of-use charge. There also is unpredictable congestion for which a congestion charge cannot be calculated in advance. In this case it is possible to imagine a congestion charge on a packet that is computed after it has been transmitted: each queue traversed by the packet calculates a charge proportional to the cumulative delay imposed by this packet on the other packets. If such a charge were instituted, a user could be charged for having his packets placed ahead of other packets in a queue.

We should not confuse usage and congestion charges. A usage charge reflects the cost of network resources that are being used. A congestion charge is a means to give network access to more valuable connections and to deny access to less valuable connections. Congestion charges reflect overall network demand and bear no direct relation to network cost. Of course, a large congestion charge is an indication of large demand, suggesting an opportunity for profit by investing in increased network capacity.

Finally, a network may provide different qualities of service, some of which require more network resources than others. The *quality* charge reflects this difference in resource use. Telephone networks and data networks today typically provide only one service quality, so quality charges are uncommon. However, with the growth of high-bandwidth applications that require guaranteed service quality (e.g., guaranteed delay bounds), services will be differentiated by quality, and quality charges will become commonplace. ATM networks are expected to provide different service quality. (The postal system charges differently for overnight, first-class, and bulk-mail delivery. This is an example of quality charges.)

In summary, economic theory suggests that a network user should pay a four-part bill comprising a fixed charge for the fixed network costs, a usage charge equal to the cost of resources used, a congestion charge that limits less-valued connections, and a quality charge for additional resources needed for higher-quality service. Practice, however, does not strictly follow theory.

### 10.3.3 Charges in Practice

Telephone companies have the most elaborate pricing schemes, including access fees, usage charges that depend on distance and call duration, and congestion charges that are approximated by time-of-use charges. Often there are quantity discounts for both fixed and usage charges, reflecting the economies of scale. In the United States the system of charges is closer to what the theory prescribes. However, it is not ideal. For example, because they are farther away from the switch, subscribers in lightly populated rural areas pay more for the local loop than those in densely populated urban areas. Nevertheless, the access charges are the same for rural and urban subscribers.

Data network charges are unsophisticated, by contrast. A local area network usually belongs to a single enterprise that provides free access to its members. Because the enterprise bears all of the network costs, there is little incentive to charge users, and network costs are part of the enterprise's "overhead" costs. However, as the use and cost of LANs have increased, a fixed, internal charge per host or per user or department is often imposed to recover the cost. This is likely to be the case in enterprises that have large network costs and a distinct department that acquires network assets and provides network services. An additional benefit from instituting such an internal charge is that it elicits information about the value members place on network services. That information can be used to make decisions about network expansion.

For a flat rate, millions of subscribers receive Internet access and additional services such as access to databases, chat rooms, and directories. As we have seen, the absence of any usage-based charges makes this an inefficient way of providing service.

Some network services such as SMDS and Frame Relay include a form of usage charge. In SMDS, for example, users subscribe to a service parameterized in terms of the sustained information rate, the maximum burst size, and the maximum number of interleaved messages (see section 3.7), and the charge for the service depends on these parameters. Since these parameters reflect the network resources devoted to providing this service, the charge includes a usage component. ISDN charges also include a usage component.

We have already noted that quality charges are uncommon today. However, the Internet protocol does provide for distinguishing among different types of datagrams using the TOS field (section 4.3). This example suggests that one way of achieving service quality differentiation is through priority of service, which guarantees preferential treatment (but not a guaranteed delay or loss bound). In principle, ATM will permit the most sophisticated forms of charges and service definition.

### 10.3.4 Vulnerability of the Internet

Until 1990, the Internet was heavily subsidized, and most users belonged to universities and research institutions. The Internet was rarely congested. When congestion did occur, hosts exercised flow control.

Internet traffic grew because commercial Internet service providers extended the user group beyond the academic and research communities and because of the popularity of applications such as the WWW. (More than 90% of Internet traffic is now under the *com* domain.) This growth has made the Internet vulnerable and has exposed the difficulty in serving applications that need guaranteed service quality.

The Internet provides a single service of uncertain quality—best-effort service. The network accepts all connections and tries to deliver the data packets. There is no admission control, and flow control is left to hosts. When a router's buffers become full, all connections through the router suffer packet loss or increased queuing delay. Hosts detect this condition because of retransmission timeouts. They are then expected to reduce their window size, and hence their packet rate. However, there is no requirement that hosts adopt such a responsible policy. Such a requirement would be very difficult to enforce. A selfish user may deliberately decide not to exercise flow control. Indeed, if others act responsibly and reduce their packet rate, the selfish user will receive a greater share of network bandwidth and buffers. This perverse incentive further encourages abuse. The resulting congestion causes retransmission, reducing network throughput and inflicting poorer service quality on all.

An Internet service provider faces the same perverse incentive. The typical provider has access to the Internet of a certain capacity (link speed and router or gateway capacity). The provider gives users access for a fixed charge. To maximize profits, the provider has the incentive to give access to as many users as possible, since there is no service quality requirement. The service received by each user (measured by delay or loss) will of course deteriorate as the number of users increases.

In a best-effort service network, everyone receives the same service quality, measured, say, in terms of delay. This quality depends on the total network load. Since there is no admission or flow control, the load is unpredictable, and so there can be no guaranteed bound on the delay. The network thus cannot meet the needs of applications that require such delay bounds. More generally, the network poorly serves connections that need a predictable service quality. In Chapter 4 we presented proposals to upgrade the Internet protocols

that seek to provide QoS guarantees. If these proposals are implemented, the vulnerability of the Internet will be reduced.

Because the Internet is a datagram network, it lacks the control functions and associated protocols (like ATM networks) that are needed in order to provide different, guaranteed levels of service quality. It may be possible, nevertheless, to provide an indirect form of control by instituting a proper pricing system, including both usage and congestion charges. We now present a design for such a system. We will then assess the kinds of control that can be exercised with such a pricing system.

## 10.4

### A BILLING AND PROVISIONING SYSTEM FOR INTERNET CONNECTIONS

A billing system for usage charges must meter the traffic. If there is a congestion charge that depends on real-time changes in the network state, the billing system must monitor the network state and give real-time price feedback to users. If the network is able to provide variable quality service, the system must also be able to provision that service.

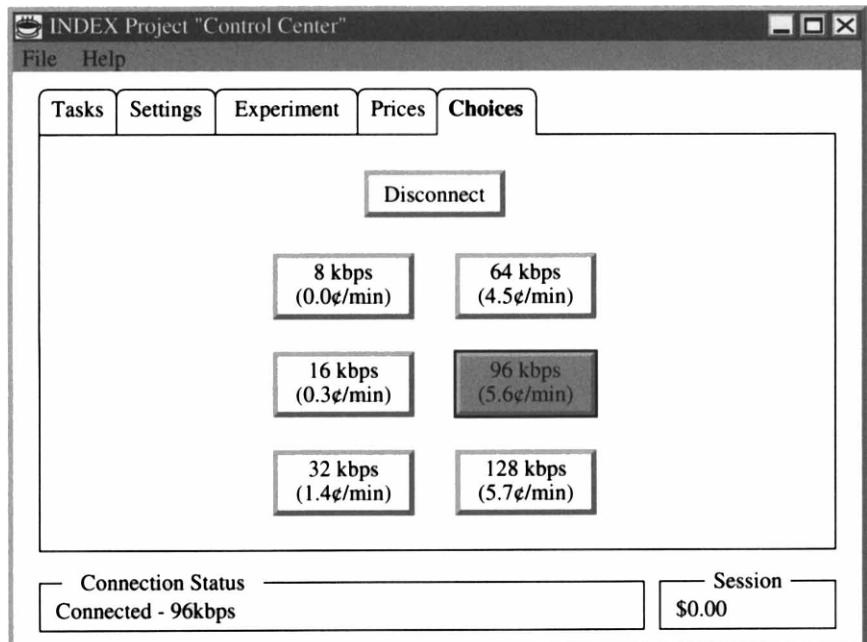
Thus, a practical billing system must have the following features:

- ◆ *No changes to existing Internet protocols and applications.* Because the installed base of hosts, routers, and user software is huge, the billing system must work without requiring changes to existing Internet protocols and widely used applications such as WWW, FTP, and e-mail.
- ◆ *User involvement.* In order to bill individual users, the system must determine the user's identity, explain the charges, and obtain approval for those charges in a secure manner.
- ◆ *On-line reporting of network usage.* In order to institute a congestion charge, the system must collect and report in real time aggregate network usage data so that the appropriate congestion charge can be calculated.
- ◆ *Sharing of information and resources.* Applications like the WWW encourage the sharing of information and resources among remote sites. Billing systems should be able to cooperate and identify users and bill them accordingly.

We will describe a billing and provisioning system developed for INDEX (Internet Demand Experiment). We will also present some early analyses of the data. INDEX is a market trial and a technology trial. The market trial involves 70 subjects affiliated with the University of California at Berkeley. INDEX customers are offered a series of service plans, each lasting eight weeks. Each service plan consists of a menu of quality-price choices. A customer can instantaneously select one of these choices. We first describe how a customer makes a choice, followed by some analyses of the data. We then describe some features of the technology.

#### 10.4.1 User Experience

After a customer connects to the INDEX network, she is presented with the choice menu. Figure 10.6 is a screen shot of the menu for one service plan. Under this plan, the customer can instantaneously select one of the offered



10.6

FIGURE

The menu offers a choice of access speeds, each priced per minute of connect time.

speeds, each priced per minute of connect time. The prices range from 0 cents per minute for 8 Kbps service to 5.7 cents per minute for 128 Kbps. (To better estimate customer demand, different customers face different randomly selected prices for the same quality choices.)

The customer indicates her choice by clicking on one of the buttons. The button is highlighted, and the status bar on the bottom indicates the current connection. The customer toggles the meter on the bottom right to find out her expenditure for the current session, for the day, and for the month to date. This meter is updated each minute so the user receives an indication in real time of the value of the network resources she is consuming. At any time the user can obtain on-line a detailed account of her expenditures.

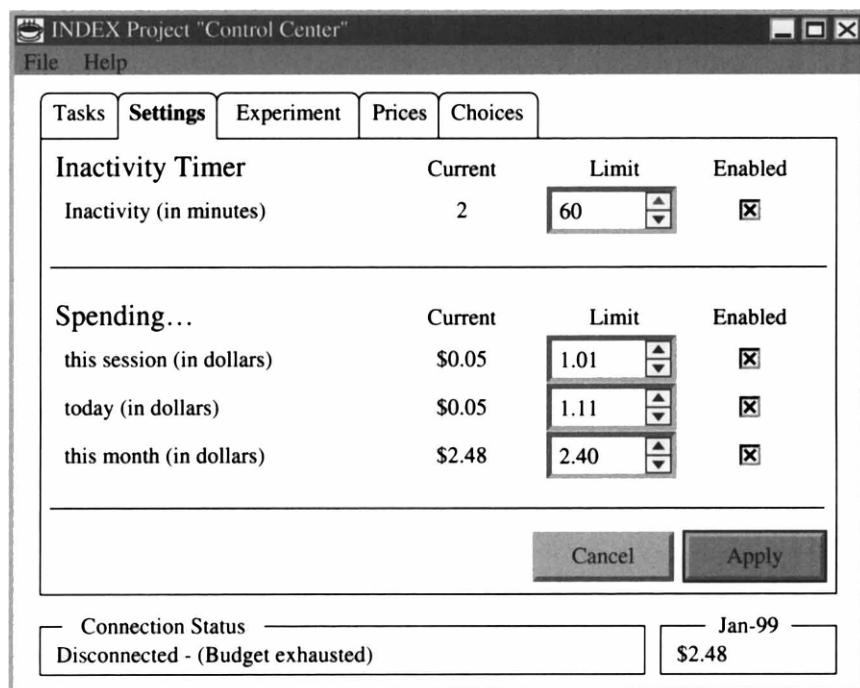
The choice that consumers make under this service plan indicates how much they value different speeds. Other service plans measure different dimensions of quality and pricing structure. These include: separate selection of the upstream and downstream bit rates, pricing by volume, combination of peak rate per minute pricing and volume pricing, and weekly flat-rate charge depending on speed.

One argument against the introduction of usage-based pricing is that a customer might be reluctant to subscribe to such a service because of the potential to run up a large bill. The INDEX customer can limit her expenditure by using the settings panel shown in Figure 10.7. This allows one to set limits on expenditure by month, day, and session. When any of these limits is exceeded, the customer's service is disconnected and the settings panel highlights the exceeded limit. The user can then raise the limit or stop consuming further service.

#### 10.4.2 Demand for Variable Quality

As we explained in section 10.2, flat-rate pricing prevents ISPs from offering differentiated quality service. What ISPs offer instead is a multi-tiered scheme in which different access speeds are offered at different flat-rate charges. Under this scheme, a customer must select one of the access speeds, but she cannot switch from one speed to another. The monthly charge for DSL access might vary between \$40 and \$200 depending on the speed. This is inefficient on several counts.

First, the difference in charge is much greater than the difference in fixed costs between DSL lines of different speeds. (In fact the DSL modems for different speeds are identical.) The different fixed charges are a form of price



**10.7**  
**FIGURE**

The settings panel allows the user to control her expenditure.

discrimination. Their purpose is to segment subscribers so that they can be charged different amounts. Second, since subscribers face no usage charge, they will waste communication resources. The waste increases with access speed. Third, subscribers who want to use higher-speed service for a limited amount of time may be unwilling to be locked into a higher-speed tier.

Both providers and subscribers lose from multi-tiered flat-rate pricing. Providers lose because they exclude low-usage subscribers from the higher-speed service. Subscribers lose because they cannot obtain the limited amount of high-speed service for which they are willing to pay.

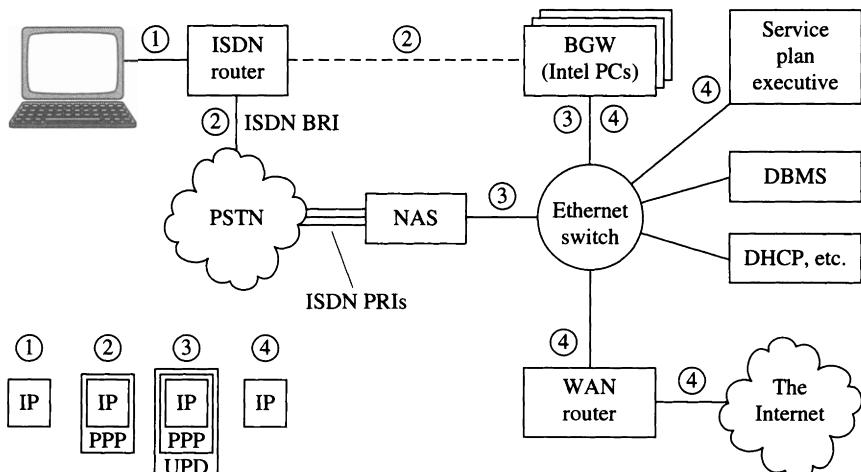
The variable-speed service plan of Figure 10.6 allows INDEX users to switch between speeds with no friction. Thus the choice users make under this service plan can be used to gauge the potential loss from multi-tiered flat-rate pricing. INDEX customers on average used three different speeds per week. Moreover, their usage of a particular speed was very sensitive both to the price of that speed as well as to the price of the adjacent speed, as seen from the estimated demand equation (10.2). This shows the very high price sensitivity

of consumers and suggests that the loss from multi-tiered flat-rate charges for variable quality to providers and consumers is large.

### 10.4.3 The INDEX Billing and Provisioning System

We describe this system by first explaining how IP packets are transferred between INDEX subscribers and the rest of the Internet. We then explain accounting and provision of service quality.

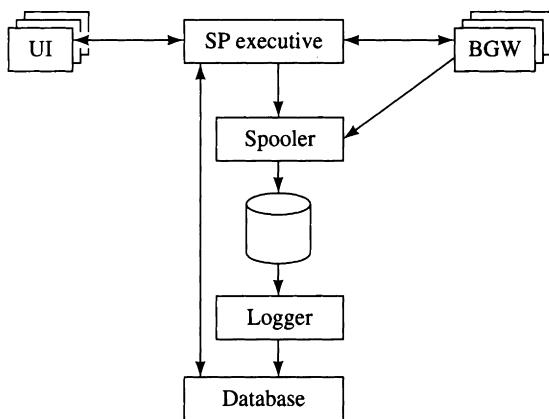
Figure 10.8 depicts the INDEX physical network. A subscriber's computer is connected via an Ethernet interface to an ISDN router with a PPP connection to the INDEX Network Access System (NAS). Traffic from many users is aggregated by the NAS. The NAS tunnels the PPP connections to the billing gateway (BGW). In this way, the network establishes a distinct logical link between each user and the BGW. In the case of the INDEX ISDN experiment, the NAS is a router that supports many ISDN interfaces. For access via CATV network, the CMTS (cable modem termination system) serves as the NAS. For DSL access, the DSM (DSL multiplexer) serves this function. The BGW routes IP packets between a user and the WAN.



10.8

FIGURE

INDEX physical network. A PPP connection over ISDN establishes a logical link between a user's computer and the billing gateway.



**10.9**  
**FIGURE**

Control packets between the user and the network operations center provide instructions for accounting and service provision.

Since the BGW is in the data path between an INDEX user and the WAN, it collects statistics such as byte counts and connection records that are to be used for accounting.

In addition to the data packets between the user and the WAN, there also are control or signaling packets between the user and the INDEX network operating center (NOC). The relations between the user and the NOC are depicted in Figure 10.9.

The left side of Figure 10.9 shows the user interface (UI), one per user. The user indicates his choice by clicking the appropriate button on the service plan window (see Figure 10.6). The Service Plan Executive observes the choice and interprets it in terms of (1) instructions to the BGW for service provision as explained below and (2) a formula that calculates the user's expenditure each minute from the statistics collected by the BGW. (For example, in case of a volume charge, the price is multiplied by the byte count received from the BGW.) This is recorded in the database and is used to update the expenditure meters on the user interface.

The BGW emulates a network of parameterized leaky buckets, one per user. The instruction from the SP Executive to the BGW specifies the parameters to match the service choice made by the user. Each leaky bucket is specified by three parameters: the sustained rate, the size of the bucket, and a small buffer for the difference between the 128 Kbps ISDN line rate and the sustained rate. For example, if the user selects 64 Kbps, the leaky bucket limits the sustained rate to 64 Kbps.

Service over DSL access may be provisioned in the same way using leaky buckets. In CATV access, the service choice may be provisioned by limiting the number of reservation slots given to a user for upstream data, whereas downstream rates may be limited by using leaky buckets.

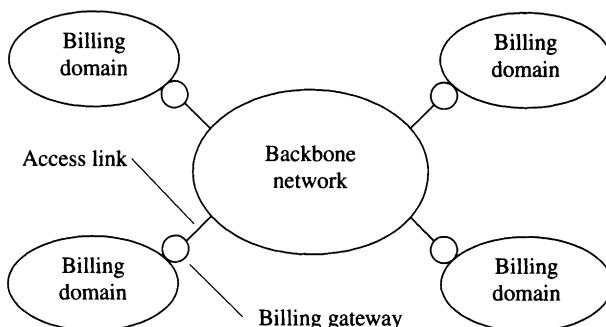
#### 10.4.4 Flexibility of INDEX Pricing and Provisioning

The INDEX pricing system is flexible: it permits pricing by combinations of flat rate, connect time, volume, peak capacity, and time of day. A certain amount of congestion pricing is possible as well. If network state can be estimated, the prices presented to the user can be a function of that state.

INDEX provisions service by limiting access. If necessary, the system can exercise admission control: a user's choice may be denied, or the service choices presented to the user may depend on the network state.

The major limitation of INDEX is that it cannot guarantee end-to-end service quality, because only access is provisioned. A distributed version of INDEX could be envisioned for the purpose of end-to-end quality. The idea is depicted in Figure 10.10.

The WAN is divided into a set of billing domains, each with an access link controlled by a billing gateway. When a user requests a particular end-to-end service, the billing gateway attempts to provision a route (using a protocol such as RSVP). An alternative approach is possible if policy-based routing is adopted in the backbone network. A quality choice is then mapped into a flow that



10.10

FIGURE

The Internet is modeled as a set of billing domains with access links controlled by a billing gateway.

receives the appropriate bandwidth or priority along its route. Neither of these approaches can assure full guarantee. What either of them can accomplish is to provide feedback to the user in the form of charges that reflect the value of the resources in the network consumed by the user. This would improve the total benefit to users of communication services. In the next section we study this in a simple context.

## 10.5 PRICING A SINGLE RESOURCE

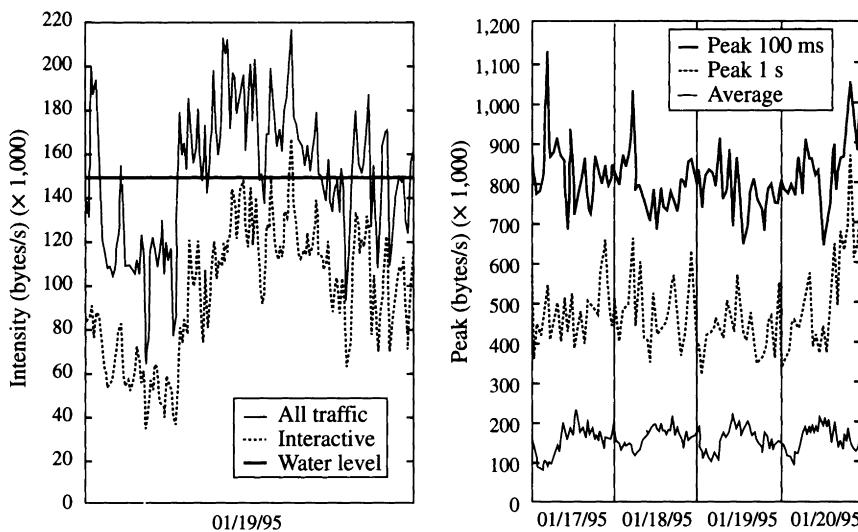
The link between a billing domain or one large organization and the Internet backbone is an expensive resource. Its cost increases as the link capacity is increased, but the service to users will improve, too. Thus there is a need to balance service benefits against capacity cost.

Figure 10.11 shows a trace of WAN traffic from the Berkeley campus obtained during an experiment referred to earlier. In the trace, SMTP, NNTP, and FTP account for 11%, 22%, and 47% of all bytes transferred, respectively. If we take the 33% of e-mail and bulletin-board traffic and spread it out over the 24 hours of the trace duration, the total amount of traffic that lies above the average rate is reduced by about 30%, as shown in the left panel of Figure 10.11. The panel on the right shows that by buffering traffic for 1 s, the 100-ms peak traffic rate is reduced by 40% (compare top and middle graphs), and by buffering for 20 min, the peak is reduced by 60% (compare middle and lowest graphs).

Assume for now that the capacity of the link between the Berkeley campus and the WAN is fixed. How can one maximize user benefits? The data shows that if we divide the traffic between delay-insensitive (e-mail and bulletin-board services) and delay-sensitive (the rest) traffic, there are two ways to reduce peak capacity and accommodate more traffic. First, by shifting delay-insensitive traffic, perhaps by several hours, to low utilization periods during the day, the peak rate can be decreased by 20%. Second, by buffering delay-sensitive traffic for a very short time on the order of a second, one can reduce the peak rate by 40%.

The shifting over time of these two types of traffic can be accomplished by administrative procedures, or by a system of time of use and congestion charges.

The administrative procedures would classify some traffic types as delay-insensitive and restrict their transmission to low utilization periods of the day. The remaining, delay-sensitive traffic would be buffered for short periods of time (on the order of 1 s). This approach is easy to implement, but it has



10.11

FIGURE

Reduction in peak traffic rate by spreading e-mail and bulletin-board traffic uniformly over the day (left); peak traffic rate measured over 0.1-s and 1-s intervals and 20-minute average (right).

two defects. It makes the determination of which traffic is delay-insensitive a matter of bureaucratic choice. Such choice is likely to be crude. For example, although e-mail typically is delay-insensitive, some e-mail connections may be urgent, but an administrative procedure cannot recognize these urgent messages. Second, the administrative classification will have to be extended over time to include new applications as they develop. This extension is bound to be arbitrary, since there is no accurate way to predict the purposes for which the new applications will be used.

A much better way to shift traffic over time is to allow users themselves to make the choice. In an ideal world, users would be altruistic and voluntarily transmit their delay-insensitive traffic during periods of low demand. But altruism is unreliable. A more reliable mechanism is through a system of prices. If users are charged a lower (or zero) rate at night, then they will have an incentive to shift their delay-insensitive traffic to those periods. For delay-sensitive traffic, a congestion charge could be imposed to keep down the peak demand. We will next study these two price mechanisms.

It is useful to keep in mind that our discussion applies to any situation in which a single resource is shared by many users. This resource could be an access link, a disk system, a pool of computers, or a highway.

### 10.5.1 Usage-Based Prices

We develop an economic model that will focus our discussion on usage-based pricing. The model is built by specifying three elements: the users' demand for service; the network capacity, that is, the amount of service that the network can supply; and the interaction between demand and supply through prices.

We consider a single service (say, best-effort service) that is differentiated by time of day. We divide the day into periods denoted by  $t = 1, \dots, T$ . For notational simplicity, consider only two periods:  $t = 1$  is the "peak" period, and  $t = 2$  is the "off-peak" period. We consider a population of potential users indexed by  $i = 1, \dots, I$ .

Consider a user's decision regarding one connection, say, to send e-mail or to browse through a WWW site. The user must choose both the period ( $t = 1$  or 2) and the amount of traffic to transmit (measured in bytes, say). We model the user's preferences by the *utility function*

$$u_t(x) = u(x) - d_t x, \quad x \geq 0, \quad t = 1, 2.$$

Here  $x$  is the amount of traffic,  $u(x)$  is the benefit (measured in dollars) that the user derives from sending  $x$ , and  $d_t x$  is the loss or benefit reduction (also measured in dollars) suffered from sending  $x$  in period  $t$ . Typically,  $d_1 < d_2$ , which is to say, most users prefer sending the traffic during the peak period.

Suppose the price for sending 1 byte in period  $t$  is  $p_t$ . If the user selects period  $t$ , she will decide to transmit a message of the size that will maximize her net benefit, that is, she will solve the problem

$$\max u(x) - d_t x - p_t x.$$

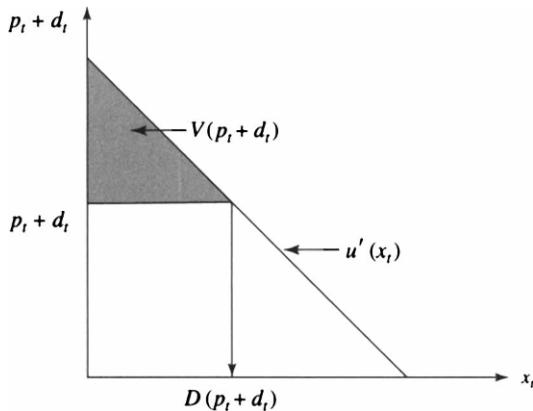
The optimum message size,  $x_t$ , is given by setting to zero the derivative with respect to  $x$ :

$$\frac{\partial}{\partial x} [u(x) - d_t x - p_t x] = 0 \text{ or } u'(x_t) = p_t + d_t. \quad (10.3)$$

We denote the net benefit she will derive from this transaction by

$$V(p_t + d_t) = u(x_t) - (p_t + d_t)x_t. \quad (10.4)$$

Here,  $u'(x) := \partial u / \partial x$  is the downward sloping curve in Figure 10.12;  $x_t$  is the size where this curve intersects the line through  $p_t + d_t$ ; and  $V(p_t + d_t)$  is the shaded area between the curve and the line. We can solve (10.3) to obtain her demand as a function of  $(p_t + d_t)$ . We write this as  $x_t = D(p_t + d_t)$ . In terms of



10.12

FIGURE

A user's demand curve  $D(p_t + d_t)$  is obtained from the marginal utility  $u'(x_t)$ . The shaded area is the user's surplus  $V(p_t + d_t)$ .

Figure 10.12,  $D(p_t + d_t)$  is simply the downward sloping curve  $u'(x)$  expressed as a function of the ordinate or  $y$  coordinate. Note that  $D$  decreases as  $p_t$  increases.

Thus, if the user decides to transmit in period 1, her benefit is  $V(p_1 + d_1)$ , and if she decides to transmit in period 2, her benefit is  $V(p_2 + d_2)$ . Clearly, she will choose the period that yields a larger benefit, that is, she will choose period 1 if  $p_1 + d_1 < p_2 + d_2$ , and she will choose period 2 otherwise.

Let us now consider an arbitrary user  $i$ , whose utility function is given by  $u_t^i(x) = u^i(x) - d_t^i x$ . This user will select period 1 if  $p_1 + d_1^i < p_2 + d_2^i$  and transmit  $D^i(p_1 + d_1^i)$ ; otherwise he will select period 2 and transmit  $D^i(p_2 + d_2^i)$ . Thus, depending on the relative price difference  $p_1 - p_2$  and the relative urgency of their connection  $d_2^i - d_1^i$ , users will segment themselves into two subsets,  $I_1$  and  $I_2$ :

$$I_1 = \{i | p_1 - p_2 < d_2^i - d_1^i\}, \quad I_2 = \{i | p_1 - p_2 \geq d_2^i - d_1^i\}. \quad (10.5)$$

Users in group  $I_1$  will transmit during period 1; those in group  $I_2$  will transmit during period 2. The resulting *aggregate* traffic demand in each period will be

$$D_1(p_1, p_2) = \sum_{i \in I_1} D^i(p_1 + d_1^i), \quad D_2(p_1, p_2) = \sum_{i \in I_2} D^i(p_2 + d_2^i).$$

The results are intuitive: demand will shift from one period to another as the first period becomes relatively more expensive; moreover, if both prices,  $p_1$  and  $p_2$ , are increased, keeping  $p_1 - p_2$  fixed, then both  $D_1$  and  $D_2$  will decrease.

Note that once the network sets the two prices, users will themselves decide which period to use and how much traffic to send in each connection. Thus a user will send urgent e-mail during the expensive period and nonurgent e-mail during the cheaper period.

Now that we understand how demand is affected by prices, we turn to a more difficult question: how should the network set the prices  $p_1, p_2$ ? The first thing to observe is that the prices should be such that the traffic demand in each period does not exceed the "supply," that is, the amount of traffic that can be transmitted over the billed link during each period. Let  $C_t$  be the total amount of traffic (in bytes) that can be transmitted in period  $t = 1, 2$ . Then, one requirement on prices is

$$D_t(p_1, p_2) \leq C_t, \quad t = 1, 2.$$

However, this still leaves a large range of choice for the prices.

In order to determine the optimum prices, we shall temporarily adopt the role of an omniscient planner who knows the utility function of each user and who will act on the user's behalf to determine for each connection which period to use and how much traffic to generate. The planner will make these choices in a way that maximizes total benefit, that is, the planner will solve the maximization problem

$$\max \sum_{i \in I_1} [u^i(x_1^i) - d_1^i x_1^i] + \sum_{i \in I_2} [u^i(x_2^i) - d_2^i x_2^i] \quad (10.6)$$

$$\text{subject to } \sum_{i \in I_1} x_1^i \leq C_1, \quad \sum_{i \in I_2} x_2^i \leq C_2. \quad (10.7)$$

The planner determines which connections  $i$  to assign to the two periods, the sets  $I_1, I_2$ , and how much data  $x_t^i$  to transmit, subject to the requirement (10.7) that the total traffic in each period be less than the available capacity. The planner will do this in a way that maximizes the benefits summed over all users (10.6). The solution to this is called the *social welfare optimum*. The following important result of economics characterizes this optimum.

**Theorem 10.5.1** The optimum is characterized by two prices  $p_1, p_2$  such that equations (10.3) and (10.5) hold and, moreover, for each period  $t$

$$\frac{\partial u^i}{\partial x_t^i} = p_t + d_t^i, \quad \text{for all } i \in I_t \quad (10.8)$$

$$I_1 = \{i | p_1 - p_2 < d_2^i - d_1^i\}, \quad I_2 = \{i | p_1 - p_2 \geq d_2^i - d_1^i\} \quad (10.9)$$

$$\sum_{i \in I_t} x_t^i = C_t. \quad (10.10)$$

The result says that the social welfare optimum can be achieved through a market mechanism in which the network optimally sets two prices  $p_1$  and  $p_2$ . Each user then selects the period to transmit and how much traffic to generate so as to maximize her own benefit (equations (10.8), (10.9)). Lastly, the optimal price for a period is such that the demand in each period equals that period's capacity (equation (10.10)).

The result suggests a practical implementation of an adaptive price-setting rule to find the optimum prices. The network begins with an arbitrary pair of prices  $(p_1, p_2)$  and measures the aggregate demand in response to these prices. If the demand in a period  $t$  is lower than its capacity  $C_t$ , the network lowers the price  $p_t$ ; if it exceeds capacity,  $p_t$  is increased.

For future reference, let us note that the network revenue generated by this usage-based pricing equals

$$R_{usage} = \sum_{t=1}^2 \sum_{i \in I_t} p_t x_t^i = \sum_{t=1}^2 p_t C_t. \quad (10.11)$$

The second equality follows from (10.10).

### 10.5.2 Congestion Prices

We now consider delay-sensitive traffic. By definition, users suffer significant reduction in benefit if this traffic is delayed even by a fraction of a second. This delay is queuing delay. It occurs when the *rate*  $\Lambda$  of total user traffic (measured, say, in bytes/s) approaches the link capacity  $M$  (bytes/s).

Queuing delay is very different from the situation considered previously, where we compared the total traffic demand  $D_t$  (bytes) over period  $t$  with the number of bytes ( $C_t$ ) that can be transmitted during that period. ( $C_t$  is the product of  $M$  and the duration of period  $t$ .)

As before, we consider a population of users, indexed by  $i = 1, \dots, I$ . The benefit that user  $i$  derives by transmitting (delay-sensitive) traffic at rate  $\lambda^i$  bytes/s is given by the utility function

$$u^i(\lambda^i) - \gamma^i \times d \times \lambda^i.$$

Here  $u^i(\lambda^i)$  is the dollar value of transmitting  $\lambda^i$  bytes/s;  $d$  is the delay faced by each byte that is transmitted, and  $\gamma^i$  converts the delay into user  $i$ 's perceived dollar cost. Suppose that the user is charged a congestion price of  $p_c$  per unit rate. (The price  $p_c$  is the price per unit of bandwidth, its unit is dollars per byte/s. By contrast, usage price discussed previously is dollars per byte.) Then user  $i$  will choose to transmit at rate  $\lambda^i$ , which solves the following problem:

$$\max_{\lambda^i} u^i(\lambda^i) - \gamma^i d \lambda^i - p_c \lambda^i.$$

The optimum rate  $\lambda^i$  is obtained by solving the equation

$$\frac{\partial u^i}{\partial \lambda^i} = \gamma^i d + p_c. \quad (10.12)$$

Knowing  $u^i$ ,  $\gamma^i$  we can solve (10.12) and obtain user  $i$ 's demand for bandwidth  $\lambda^i = D^i(p_c)$  as a function of the congestion price  $p_c$ . Let us also note the *aggregate* bandwidth demand

$$\Lambda = \sum_i \lambda^i = D(p_c) = \sum_i D^i(p_c).$$

We now consider the problem of the omniscient planner who chooses  $\lambda^i$  on behalf of user  $i$  so as to maximize the total benefit

$$\sum_i [u^i(\lambda^i) - \gamma^i d \lambda^i].$$

The queuing delay  $d$  is a function of the total traffic rate  $\Lambda = \sum \lambda^i$  and the link capacity  $M$ , which we write as  $d = f(\Lambda, M)$ . So the planner's problem is

$$\max \sum_i u^i(\lambda^i) - f(\Lambda, M) \sum_i \gamma^i \lambda^i.$$

The optimum values of  $\lambda^i$  are obtained by solving the equations

$$\begin{aligned} \frac{\partial u^i}{\partial \lambda^i} &= \gamma^i f(\Lambda, M) + \frac{\partial f}{\partial \Lambda}(\Lambda, M) \sum_j \gamma^j \lambda^j \\ &= \gamma^i d + \frac{\partial f}{\partial \Lambda}(\Lambda, M) \sum_j \gamma^j \lambda^j, \quad i = 1, \dots, I. \end{aligned} \quad (10.13)$$

The right-hand side of (10.13) is the sum of two terms. The first term is the cost of delay directly suffered by user  $i$ . The second term is the increase in the

delay cost suffered by all users due to a unit increase in user  $i$ 's traffic rate. This term is therefore called the *congestion cost*. Comparing (10.12) and (10.13) we see that the welfare optimum is achieved if the congestion price charged to each user equals the congestion cost.

**Theorem 10.5.2** The optimum is achieved by charging each user the congestion price of

$$p_c = \frac{\partial f}{\partial \Lambda}(\Lambda, M) \sum_j \gamma^j \lambda^j \quad (10.14)$$

per bytes/s for one unit of time, say one hour.

As an example, we compute the congestion price for an M/M/1 queuing model. The delay is given by

$$f(\Lambda, M) = \frac{1}{M} \frac{\Lambda}{M - \Lambda},$$

from which the congestion price can be calculated to be

$$\frac{\partial f}{\partial \Lambda} \sum_i \gamma^i \lambda^i = \frac{\sum_i \gamma^i \lambda^i}{M^2} \left[ \frac{\rho}{1 - \rho} + \frac{\rho^2}{(1 - \rho)^2} \right],$$

where  $\rho := \Lambda/M$  is the utilization. We see that as  $\rho$  approaches 1, the congestion price increases rapidly. Facing such a large price, users will reduce their traffic rate.

The Berkeley campus data showed that the total traffic rate varies widely from one second to the next. The congestion price will vary equally rapidly. It is not practical for the network to communicate such a price variation on a second by second basis and for users to react to such rapid price variation. Thus schemes to implement congestion pricing must proceed indirectly. We describe one such scheme, based on the notion of a *reservation price*.

In this scheme, whenever user  $i$  sets up a connection, she provides the network her “reservation price”  $p_i$ . ( $p_i$  is  $i$ 's maximum “willingness to pay” for traffic in that connection.) The understanding is that the network will buffer her packets until the congestion price falls below  $p_i$ , at which time her traffic will be forwarded. Thus the network maintains a list of users, ordered by their reservation prices. The network continually computes the congestion price and transmits those packets whose reservation price exceeds the congestion price. Users are charged according to the prevailing congestion price. This scheme guarantees that users would pay less than their reservation price.

The revenue (per hour) from the congestion price is

$$R_{cong} = p_c \sum_i \lambda^i = p_c \Lambda. \quad (10.15)$$

Most users would find it bewildering to face a complex pricing scheme such as congestion prices, which can shift unpredictably over time. These schemes may be proper for large corporate users who are purchasing transmission services on behalf of many individuals in the corporation. The scheme may also be sensible for network access providers. These providers may purchase services in the congestion "market" but make them available to their customers at a fixed price, larger than the average congestion price that the access provider faces. From a social viewpoint, this is a good arrangement: the access provider is bearing the full cost of congestion and absorbing the risk, whereas the individual users face no risk but pay a "risk premium" above the average cost.

### 10.5.3 Cost Recovery and Optimum Link Capacity

The revenue collected by the usage-based and congestion pricing schemes may exceed or fail to cover the cost of the billed link. To simplify the comparison between revenue and cost, we will ignore congestion prices. So total revenue is

$$R = R_{usage} = \sum_{t=1}^2 p_t C_t = \sum_{t=1}^2 p_t M L_t,$$

where  $M$  is the link capacity (bytes/s) and  $L_t$  is the duration of period  $t$ . Since the two periods  $t = 1$  and  $2$  add up to one day, this is the revenue per day.

So far we have assumed that the link capacity  $M$  is fixed. We will now suppose that  $M$  can be varied and that the daily cost of renting a link of capacity  $M$  is  $r(M) = r_{fix} + r_{var} \times M$ . Thus the cost has a fixed component  $r_{fix}$  and a variable component  $r_{var}M$  that is proportional to the capacity. (A cost structure for a link comprising a fixed term and a variable term is very common. The variable cost may not be proportional to the capacity, as we have supposed here for simplicity.) Hence the net benefit  $B(M)$  of a link of capacity  $M$  is the revenue minus cost, on

$$B(M) = \sum_{t=1}^2 p_t M L_t - r_{fix} - r_{var}M.$$

The optimal capacity is obtained by maximizing  $B(M)$ , that is, by setting  $\partial B/\partial M = 0$ . This gives

$$\sum_{t=1}^2 p_t L_t = r_{var}. \quad (10.16)$$

The left-hand side is the increase in revenue resulting from a unit increase in the link capacity, and the right-hand side is the cost of that incremental capacity.

If we select the optimal capacity, then the daily revenue will be  $r_{var}M$ , which is the variable part of the capacity cost. The fixed cost  $r_{fix}$  will not be covered by the usage-based price scheme. If it is not possible to cover this through a subsidy, then an alternative is to impose a fixed charge on users. That is, any user who wants access to the billed link will have to pay a daily subscription charge, regardless of how much traffic the user will transmit.

Suppose this fixed charge is  $p_{fix}$ . If there are  $I$  users in all, then this fixed charge should be  $p_{fix} = r_{fix}/I$ . Let us study the impact of this fixed charge on a user who decides to transmit  $x_t$  bytes during period  $t$ . This user will now pay  $p_{fix} + p_t x_t$ . The net benefit she now derives is given by (compare (10.4))

$$V = u(x_t) - (p_t + d_t)x_t - p_{fix}.$$

There are two cases to consider.

If this is a user with a large demand  $x_t$ , then  $V > 0$ , despite the fixed cost. This user will pay the fixed charge and generate the same traffic as before. On the other hand, if this is a user with a small demand  $x_t$ , then  $V < 0$ , that is, this user finds the fixed cost to be so large that the net benefit is negative. Thus low-demand users will refuse to subscribe. This is undesirable, since the link capacity is large enough to accommodate them. We conclude that, ideally, the fixed charge should be levied only on the high-demand users who would continue to subscribe. However, it may not be possible in practice to discriminate in this way between high-demand and low-demand users.

We can summarize the discussion in this section. We considered a single resource, such as the billed link for the Berkeley campus. We studied how access to this link should be charged by a three-part pricing scheme: a usage-based price that varies by time of day and encourages users to shift their delay-insensitive traffic to periods with a low demand; a congestion price that encourages users to reduce their traffic rate when the resource is congested; and a fixed charge to recover fixed costs, ideally imposed only on high-demand users. In the next section we consider multiple resources.

## 10.6 PRICING FOR ATM SERVICES

In the preceding section we discussed the pricing of a single service that was provided using a single resource. The service demand was measured either by the number of bytes or the rate (in bytes/s) of the user's traffic. The capacity of the resource was correspondingly measured either by the maximum number of bytes that the link could transmit in a given period or the maximum transmission rate. Because of this direct correspondence between the service that users demand and the capacity of the resource, we can think of the price either as a price per unit of service or as the rent for a portion of the resource capacity sufficient to produce that unit of service. Thus, for example, the congestion price  $p_c$  is the price to transmit traffic at a rate of 1 byte/s for one hour. It can also be regarded as the hourly rent of  $1/M$  of the link capacity of  $M$  bytes/s.

It becomes essential to distinguish between prices for services and rents for resources in ATM networks, because different resources are used to provide many different transmission services. We will focus on two sets of resources: the capacities of the different links and the buffers associated with each link. We will be concerned with services that transmit traffic with certain burstiness characteristics within a certain delay. Thus different services offered by the network will be distinguished by burstiness parameters and delay bounds.

Users and the network service provider enter into a contract. The contract specifies the burstiness parameters, the delay, and the price. It obligates the user to make sure that his traffic will conform to the burstiness parameters. It obligates the network to transfer conforming traffic within the specified delay, in exchange for the specified price. (We saw an example of such a contract in the form of GCRA in section 8.4.2.)

The network (service provider) meets its obligation by (1) selecting a route and (2) by *reserving* bandwidth and buffers in each link along the route in amounts sufficient to meet the delay requirement. The network can choose different routes, and it can dedicate different amounts of resources to meet the requirement. The network will select those combinations of routes and resources that will maximize its revenue. We will first describe a model that allows us to formulate the question of revenue maximization. A variant of this formulation will address the question of optimum prices for services. This will be the counterpart of Theorem 10.5.1. Finally, we will suggest an alternative formulation in which users directly rent resources—buffers and bandwidth—from the network and decide how to satisfy their own service requirements.

### 10.6.1 A Model of ATM Resources and Services

We consider ATM networks comprising a set of links,  $L$ , interconnected by switches. Suppose that these are output buffered switches, with one buffer per link. Then each link  $l \in L$  is characterized by its transmission capacity of  $C_l$  (ATM) cells per second (or cps), and its buffer size of  $B_l$  cells. The resources of the network as a whole are given by the set of pairs

$$\{(C_l, B_l) \mid l \in L\}. \quad (10.17)$$

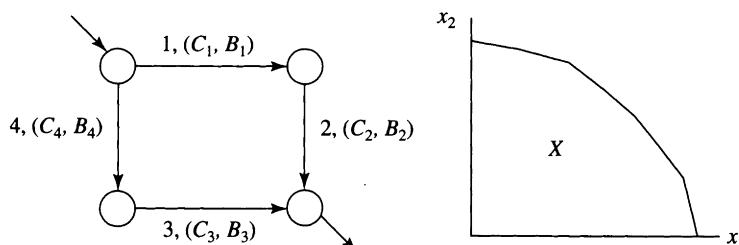
Figure 10.13 gives a model of a network with four links.

We suppose that the network provider has selected a set of routes  $R$ . Each route  $r \in R$  designates a set of links that comprises this route. (For example, one possible route in the network of Figure 10.13 is  $r = \{1, 2\}$  comprising links 1 and 2.) A route could be defined by a virtual path. Lastly, we suppose that the network provider has designed a set of service provision activities  $A$ . To each activity of type  $a \in A$  there corresponds a route and bandwidth and buffer reservations on the links along the path. Formally, an *activity*  $a$  is defined by its route  $r_a \in R$  and a list of resources along the route

$$a \rightarrow \{(c_{la}, b_{la}) \mid l \in r_a\}. \quad (10.18)$$

The interpretation is that in order to carry out the activity  $a$ , the network must reserve  $c_{la}$  cps of bandwidth and  $b_{la}$  cells of buffer in each link  $l$  along the route  $r_a$ .

When a user selects a service (to be described later), the network will choose an activity type  $a$  such that the resources (10.18) reserved by it can meet



10.13

FIGURE

The network has four links; link  $i$  has bandwidth  $C_i$  and buffers of size  $B_i$ . The feasible set of activities is  $X$ .

the delay requirements of the service that the user has selected. Of course the provider can do this only if free resources are available. Thus one wants to know how many activities of different types can be accommodated by the available network resources, given by (10.17). This leads to the following definition.

It is feasible to simultaneously accommodate  $x_a$  of type  $a \in A$  provided that

$$\sum_a \epsilon_{la} c_{la} \times x_a \leq C_l, \quad l \in L, \quad (10.19)$$

$$\sum_a \epsilon_{la} b_{la} \times x_a \leq B_l, \quad l \in L. \quad (10.20)$$

Here, we define the numbers  $\epsilon_{la} = 1$  or 0 depending on whether link  $l$  belongs or does not belong to route  $r_a$ . Thus the inequalities (10.19), (10.20) imply that there are enough bandwidth and buffers in each link  $l$  to meet the requirements simultaneously placed by  $x_a$  activities of type  $a$ .

Consider again the network of Figure 10.13. Suppose the resources of links 1 and 2 are, respectively,

$$C_1 = 10^6 \text{ cps}, \quad B_1 = 10^6 \text{ cells}, \quad C_2 = 0.8 \times 10^6 \text{ cps}, \quad B_2 = 2 \times 10^6 \text{ cells}.$$

(We ignore the other links.) Consider two activities  $a = 1$  and  $a = 2$ , both designating the same route  $\{1, 2\}$ . The resources required by activities 1 and 2 are

$$\{c_{11} = 10^3 \text{ cps}, b_{11} = 10^3 \text{ cells}, c_{21} = 10^3 \text{ cps}, b_{21} = 10^3 \text{ cells}\}, \quad (10.21)$$

$$\{c_{12} = 10^2 \text{ cps}, b_{12} = 10^2 \text{ cells}, c_{22} = 0.5 \times 10^2 \text{ cps}, b_{22} = 2.5 \times 10^2 \text{ cells}\}. \quad (10.22)$$

Then it is feasible to accommodate  $x_1$  activities of type 1 and  $x_2$  activities of type 2 if

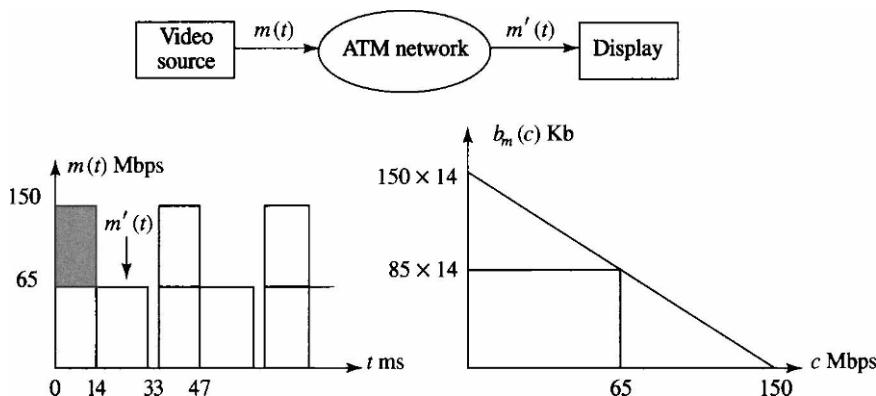
$$10^3 x_1 + 10^2 x_2 \leq 10^6, \quad (10.23)$$

$$10^3 x_1 + 0.5 \times 10^2 x_2 \leq 0.8 \times 10^6, \quad (10.24)$$

$$10^3 x_1 + 10^2 x_2 \leq 10^6, \quad (10.25)$$

$$10^3 x_1 + 2.5 \times 10^2 x_2 \leq 2 \times 10^6. \quad (10.26)$$

These inequalities represent the constraints imposed by  $C_1$ ,  $C_2$ ,  $B_1$ ,  $B_2$ , respectively. They define the feasible set  $X$  of all activities that satisfy these four inequalities.  $X$  is a convex set, because the inequalities (10.19), (10.20) are linear. (The set  $X$  in Figure 10.13 is not drawn to scale.)



10.14

FIGURE

The message  $m(t)$  is a sequence of video frames once every 33 ms. The received message is  $m'(t)$ . The burstiness curve of  $m(t)$ ,  $0 \leq t \leq 33$  ms is  $b_m(c)$ .

Note how the activity definition (10.18) transforms the set of available resources (10.17) into the feasible set  $X$ . We will now see how each activity is transformed into services that the provider can offer to users.

We consider a population of users who wish to transmit a stream of cells, whose instantaneous rate is  $m(t)$ ,  $0 \leq t \leq T$ ;  $m(t)$  is measured in cps and  $T$  is the duration of the stream in seconds. It will be convenient to call such a stream a *message*. Figure 10.14 displays an example where the user's message is a sequence of video frames. A frame contains  $512 \times 512$  8-bit pixels and is generated every 33 ms. The user requires a service that will deliver a frame every 33 ms to the display.

Suppose the message rate is as shown in the figure. If the network reserves a bandwidth of 150 Mbps (the peak rate) in every link along the route, then the received signal will be the same as  $m(t)$  (except for a constant propagation and switch processing delay). An alternative is to allocate 65 Mbps and some buffers (given by the shaded area,  $85 \times 14$  Kb) in each link; the received signal then will be  $m'(t)$  as shown. Note that both allocations meet the user's requirements. Observe that the allocations have the form of activities, that is, they reserve bandwidth and buffers in the links along a route. Clearly, if a much smaller bandwidth or fewer buffers are allocated, then the user's requirement of delivering frames cannot be met. Thus the user's message must conform to the resources allocated for it. We now specify what a conforming message is, using the idea of a burstiness curve. (We have already introduced the notion

of burstiness in Chapter 8. We repeat it here only to make the discussion self-contained.)

The burstiness curve of a message  $m(t)$ ,  $0 \leq t \leq T$ , is the function  $b_m(c)$  that gives the size of the buffer needed to transmit message  $m$  without loss over a link of capacity  $c$  cps. Clearly, the larger the rate  $c$ , the smaller the required buffer size  $b_m(c)$ . At one extreme, if  $c = 0$ , the entire message must be buffered, so

$$b_m(0) = \int_0^T m(t) dt.$$

At the other extreme, if  $c$  is larger than the peak rate of  $m$ , no buffer is needed, so

$$b_m(c) = 0, \quad c \geq \max_t m(t).$$

A useful property of the burstiness curve is that it is convex. Figure 10.14 shows the burstiness curve of one frame for the video source.

Suppose the message  $m$  is transmitted over a route  $r$  comprising links  $l \in r$ , and suppose  $c_l$  cps of bandwidth and  $b_l$  cells of buffers are reserved for this message in link  $l$ . Then this message will be transmitted without loss if in every link the reserved buffer size exceeds the burstiness curve at the reserved rate, that is, if

$$b_l \geq b_m(c_l), \quad l \in r.$$

Moreover, if  $c_{min} = \min\{c_l | l \in r\}$  is the minimum reserved bandwidth, then the total end-to-end delay suffered by the message is

$$\text{delay} = \frac{b_m(c_{min})}{c_{min}} + \text{propagation and processing delay.} \quad (10.27)$$

We can now describe the services that the network provider may offer to users. A *service* is a four-tuple  $s = (r_s, b_s(c), c_s, \delta_s)$ , where  $r_s$  is a route,  $b_s$  is a burstiness curve,  $c_s$  is the minimum transmission bandwidth, and

$$\delta_s = \frac{b_s(c_s)}{c_s} + \text{propagation and processing delay}$$

is the guaranteed delay. (Any nonnegative, convex, decreasing function  $b(c)$ ,  $c \geq 0$  is a burstiness curve.) Let  $S$  denote the set of services offered by the network. Each service  $s$  is sold at a price  $p_s$  per unit of time. (The reader will note that this definition of service is a generalization of the ATM Forum's GCRA proposal.)

If a user wishes to transmit a message  $m(t)$ ,  $0 \leq t \leq T$ , she can purchase a contract for service  $s = (r_s, b_s, c_s, \delta_s)$  for time  $T$ . She must pay  $p_s \times T$ , and the contract requires that her message be *compliant*. This means that her message must be less bursty than the service burstiness curve, or

$$b_m(c) \leq b_s(c), \quad c \geq c_s. \quad (10.28)$$

In return, the contract requires the network to transmit her message without loss, over route  $r_s$ , and with a delay not exceeding  $\delta_s$ . To fulfill its side of the contract, the network provider selects an activity  $a$  that can meet the requirements of  $s$ . This means the two routes are the same, and the resources reserved by  $a$  are sufficient, that is,

$$r_a = r_s; \text{ and } b_{la} \geq b_s(c_{la}), c_{la} \geq c_s, \quad l \in r_a. \quad (10.29)$$

Let us check that this selection is proper. Let  $c_{min} = \min\{c_l | l \in r_s\}$ . Then

$$\frac{b_m(c_{min})}{c_{min}} \leq \frac{b_m(c_s)}{c_s} \leq \frac{b_s(c_s)}{c_s}.$$

The first inequality follows since  $c_{min} \geq c_s$  by (10.29), and the second inequality follows from (10.28). Finally, using this inequality in (10.27) implies that the delay suffered by the message is less than  $\delta_s$ , as required by the contract.

## 10.6.2 Revenue Maximization

We have seen above that in order to fulfill a contract for service  $s$ , the network provider must undertake an activity  $a$  that satisfies (10.29). There may be several activities that meet this requirement. For example, we saw in Figure 10.14 that the service requirement could be met by providing peak bandwidth and no buffers or a lower bandwidth and some buffers. The network provider must consider which activity to assign to each service so as to maximize the network revenue. We formulate this revenue maximization problem.

For each service  $s$  let  $A_s$  be the subset of activities that satisfies (10.29). Let  $x_{sa}$ ,  $s \in S$ ,  $a \in A_s$  be the number of units of service  $s$  that are assigned to activities of type  $a$ . Then the number of units of service  $s$  that are sold with this assignment is

$$n_s = \sum_{a \in A_s} x_{sa}, \quad s \in S.$$

The revenue (per unit of time) from this assignment will be

$$\text{Revenue} = \sum_s p_s n_s = \sum_s \sum_{A_s} p_s x_{sa}.$$

On the other hand, the number of simultaneous activities of type  $a$  resulting from this assignment is

$$x_a = \sum_s x_{sa}, \quad a \in A,$$

where we adopt the convention that  $x_{sa} = 0$  if  $a \notin A_s$ . This assignment of services to activities must be feasible, that is, the activities  $\{x_a, a \in A\}$  must meet the constraints (10.19), (10.20).

Combining these observations, we can see that the revenue is maximized by the assignment  $\{x_{sa}, s \in S, a \in A\}$ , which solves the following problem

$$\begin{aligned} & \max \sum_{s \in S} \sum_{a \in A} p_s x_{sa} \\ \text{subject to } & \sum_s \sum_a \epsilon_{la} c_{la} \times x_{sa} \leq C_l, \quad l \in L, \\ & \sum_s \sum_a \epsilon_{la} b_{la} \times x_{sa} \leq B_l, \quad l \in L \\ & x_{sa} = 0, \quad a \notin A_s. \end{aligned}$$

This is a linear programming problem, which can be solved using standard algorithms.

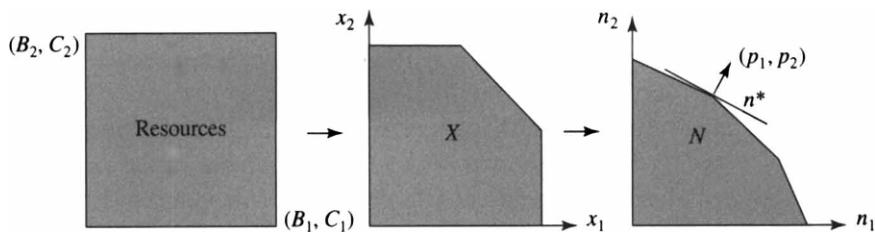
There is a useful geometric picture that relates the set  $X$  of feasible simultaneous activities and the set  $N$  of feasible service units. Let us say that it is feasible to sell simultaneously service units  $n_s$  of type  $s$ , if there is an assignment  $x_{sa}$  such that

$$n_s = \sum_{a \in A_s} x_{sa}, \quad s \in S,$$

and the corresponding activities

$$x_a = \sum_s x_{sa}, \quad a \in A,$$

are feasible, that is, these activities are in  $X$ . Let  $N$  be the set of feasible service units. Then, like  $X$ , the set  $N$  is a convex set. The maximum revenue is then given by maximizing  $\sum_s p_s n_s$  over the set  $N$ .



**10.15**  
**FIGURE**

Network resources are transformed into the set  $X$  of feasible activities, which in turn is transformed into the set  $N$  of feasible services. For prices as shown, revenue is maximized at  $n^*$ .

We summarize the two steps developed in this section. We start by modeling the ATM network as a collection of links  $L$ , with each link  $l$  being characterized by its bandwidth and buffers,  $(c_l, b_l)$ . Next, the network provider designs a set of activities  $A$ , with each activity  $a$  associated with a route and bandwidth-buffer reservation along the route. This determines the set  $X$  of feasible activities. The network provider offers for sale a set  $S$  of services and finds which activities  $A_s$  can fulfill each service  $s$ . This determines the set  $N$  of services that can be supplied for sale. Figure 10.15 illustrates the two steps: from resources to activities, and from activities to services.

## 10.7

## SUMMARY

Technical discussions on networking carried out in textbooks, in journals, or in conferences normally do not discuss economic issues, either in terms of costs or in terms of the use of prices and other market mechanisms as a form of control and resource allocation. There are obvious reasons for this. The most important networks—the telephone and CATV networks—until recently were subject to regulation that effectively set prices for their services. The most important data networks—LANs—were privately owned, and the Internet was fully subsidized and free. Equally importantly, each network offered a single, undifferentiated service, so economic issues generally have been reduced to questions of cost and scale economics and, of course, regulation.

With the rapid commercialization of the Internet with its potential to provide services that compete with the existing networks, the remarkably rapid developments in ATM and, more generally, the movement within the entire

communications industry to compete to reap the economies of service integration and network externalities, economic issues are coming to the foreground. Network engineers and computer scientists who previously were engaged in purely technological concerns are finding that economic "variables" can be used to manage the network just as much as admission and flow control.

This chapter provided a very brief introduction to a variety of microeconomic models that can be used to formulate questions of network control and performance. In several important ways the economic models build on top of the models developed in Chapters 8 and 9. We made particular use of the deterministic models of traffic. We believe that network engineers are in a particularly good position to address economic issues because of their familiarity with the technology. We hope that this introduction will encourage them to turn their attention to those economic issues.

As we mentioned in the introduction, a more complete consideration of supply factors must include the costs of networking technology. Those estimates are not available, at least in the published literature, but we can provide an appreciation of the advances that are bringing such dramatic changes in networking. The next two chapters are devoted to those advances.

## **10.8**

## **NOTES**

For a general introduction to microeconomics, see [V93]. For a wide-ranging analysis of information goods, see [SV99]. A classic analysis argued that in competition for votes, viewers, and customers, politicians, TV broadcasters, and producers tend to become like one another [H29]. The material in section 10.4 is adapted from [EMV95, E99]. The Point-to-Point Protocol (PPP) provides a standard method for transporting multiprotocol datagrams over point-to-point links (RFC 1548). For a congestion pricing scheme for TCP-like protocols see [GK98]. Section 10.5 is based on [MV95]. Section 10.6 is based on [JV91, JV94]. For Internet accounting requirements see RFC 1272.

## **10.9**

## **PROBLEMS**

1. Suppose that it costs more to provide (telephone or CATV or Internet) access in rural areas than in urban areas. Would it be more efficient to impose larger access charges on rural experiences? On what public policy

grounds should the government require the same access charge, which, in effect, implies that urban subscribers subsidize rural subscribers?

2. Suppose that wages and salaries are lower in rural than in urban areas but that network access charges are the same. Consider a business that serves its customers through the network. (For example, a mail-order business takes most of its orders over the telephone.) Do you think it more likely that such a business would be located in a rural area? Can you find any empirical evidence that might support or deny such an expectation?
3. Suppose a company sells Internet access through dial-up modems. The company has two modem pools: the fast modems have a speed of 28.8 Kbps and the slow modems have a speed of 2.4 Kbps. Subscribers can dial either modem pool. Suppose that from noon to 6 p.m., the fast pool is congested, that is, more subscribers wish to gain network access than the number of modems in the fast pool. What is your prediction about subscriber behavior during the busy period in terms of (1) their willingness to keep on dialing the fast modem pool until they get through, (2) their willingness to dial into the slow modem pool, and (3) their unwillingness to attempt access during the busy period?
4. If a company creates a unique product (software or hardware) that is considered valuable by users, the company can charge much more for this product than its cost. After some time, however, other companies will develop the same or a substitute product, and competition will drive the price to the cost of production. Can you build a model of consumer behavior and market organization that supports this? A market for a particular product is said to be contestable if the period for which the first company enjoys a monopoly is relatively short. Do you think that software product markets are likely to be more contestable or less contestable than hardware products? Why?
5. Companies A and B both make computers. A's computer has an open architecture, whereas B's has a proprietary architecture. How would you compare the prospects of these two companies?
6. This problem seeks to model the vulnerability of the Internet. Suppose  $n$  users have TCP connections with traffic rates  $x_1, \dots, x_n$  going through the same router. The queuing delay they all experience is some function  $d(x)$ , where  $x = x_1 + \dots + x_n$ .
  - (a) Since queuing delay increases with the traffic rate,  $d(x)$  must be an increasing function of  $x$ . What more can you say about the behavior of  $d$ ? Suggest a specific form of  $d$  based on the discussion of section 8.3.

- (b) Suppose the value to user  $i$  of the connection is given by  $V_i(x_i, d)$  where the function  $V_i$  is increasing in  $x_i$  and decreasing in  $d$ . Is this form of the value function plausible? Suppose the  $n$  users cooperate one another and decide to set their traffic rates  $x_i$  so as to maximize their total benefit,

$$\max_{x_1, \dots, x_n} \sum_i V_i(x_i, d).$$

Derive equations that can be solved to find the optimum rates  $\{x_i^*\}$ .

- (c) Suppose the users are noncooperative. We model noncooperative behavior by saying that user  $i$  selects  $x_i$  to maximize  $V_i(x_i, d)$ , taking  $d = \sum_{j \neq i} x_j$  as fixed. Let the resulting traffic rates be  $x_i^u$ . Do you expect  $x_i^u$  to be larger or smaller than  $x_i^*$ ? Can you support your guess by a mathematical argument?
- (d) One way of achieving the cooperative solution is through congestion pricing. Another way is through some sort of social regulation in which users who do not reduce their traffic rate during congestion are subject to some kind of social sanction. Can you think of such a regulation mechanism? Compare such social regulation with market regulation through pricing. What are the pros and cons of the two schemes? (The discussion in section 10.5 should help in answering this question.)
7. Proposals to upgrade Internet protocols were discussed in Chapter 4. The argument for an upgrade is that it will permit the Internet to support applications that require QoS guarantees. The argument against the upgrade is that it will render obsolete the large installed base of IP software around the world. (The situation is more complicated than this, but we ignore the complications.) Assess the two arguments. Clearly, users who want QoS guarantees will favor the upgrade and would be willing to bear the upgrade cost, and those who do not want such guarantees will be unwilling to bear the cost since they derive no additional benefit. Can you propose an upgrade strategy whereby the former can compensate the latter for the additional cost?
8. Analyze the additional TCP connection setup time resulting from implementing the billing system of section 10.4.
9. Propose a billing scheme for ATM connections.
10. Talk to your network manager and figure out the cost of Internet access as a function of bandwidth, number of users, total traffic, and any other parameters you find important.

11. Find out the charges for Internet access offered by three network access providers. Why are the charges different? Are they appealing to different segments of users? What costs are incurred by the access provider?
12. A small business office seeking Internet access has two options (among others). The office can install a number of modems to provide access through an ISP. Alternatively, the office can connect its computers via Ethernet and rent or lease a high-speed link (e.g., an ISDN or Frame Relay or SMDS service). Compare the two options.
13. Video stores offer to rent or sell a video of the *same* movie. Since the rental price for the video is much less than the sales price, it seems surprising that anyone would buy the video rather than rent it. Can you explain why both markets coexist? (Saying that consumers are irrational is not an explanation. Part of the answer may be discovered by finding out which movies are offered for sale.)
14. Software and books are both easy to copy, and copyright laws do not provide much protection. The pricing strategies for these two products seem quite different. The price of software has gone down steadily as manufacturers have gone for the mass market. The price of books has gone up steadily as publishers have restricted their market to libraries and businesses that are more likely to abide by copyright laws. Thus the software producers have accommodated to the ease of illegal copying by lowering prices, thereby reducing the incentive to make illegal copies, whereas publishers have limited their markets. Compare the two strategies.
15. We saw the diversity of Internet traffic on the Berkeley campus. Propose a pricing scheme that takes this diversity into account and that is simple to implement, understandable to users, and recovers costs.
16. Suppose the telephone switch on the University of California-Berkeley campus costs \$300,000 per year. Suppose 10,000 phones are served by this switch. This cost can be recovered in at least two different ways. First, one may charge a fixed fee of \$30 per month for each phone, or one may charge a usage price of  $p$  per phone call, with  $p$  adjusted to recover the switch cost. Discuss the relative merits of these two forms of charges.

# Optical Networks

In this chapter we describe the advances in optical networking that made possible the explosive growth of communication networks of the 1990s. We also point out the directions in which optical networking is likely to evolve.

The bandwidth of copper cables declines rapidly to 100 MHz over a 1-km distance, before signal regeneration is required. By contrast, an optical fiber has a bandwidth of 25,000 GHz over a distance of several tens of kilometers. That capacity is already used in several ways.

Cable TV fiber distribution networks utilize a bandwidth of about 1 GHz. Optical links have increased LAN speeds to 1 Gbps, and 100 Mbps Ethernet links to the desktop are no longer uncommon.

The capacity of telephone and data backbone networks increased by several orders of magnitude from DS-3 (45 Mbps) links in 1990 to OC-48 (2.5 Gbps) SONET fiber links in 1997. This is still a tiny fraction of the 25-THz fiber bandwidth. The reason for the limit is that electronic modulators today have a maximum speed of 2.5 Gbps. That speed will soon increase to 10 Gbps.

By 1997 commercial dense wave-division multiplexers (WDM) overcame the electronic speed limit by transporting forty 2.5-Gbps channels on the same fiber, increasing total link capacity to 100 Gbps. Companies have announced WDM products capable of carrying 160 10-Gbps channels. Thus very high-speed WDM links are being deployed.

The next wave of products will include optical cross-connects (OXCs) that allow combining links into “lightpaths” entirely in the optical domain. This is similar to how SONET paths are constructed using digital cross-connects or switches. The advantage of OXCs is that no processing is required. Moreover the lightpaths in some cases are transparent to higher-level protocols.

The next advance will occur when OXCs can be reconfigured quickly, at least on the time scale of a connection. This will augur the creation of very high bandwidth-on-demand services.

Data traffic will soon be larger than voice in backbone traffic, and this will promote transport solutions that bypass the time-division multiplexing structure of SONET. The advantages of statistical multiplexing, the flexibility of packet switching in accommodating multiple types of traffic, and the lower cost of high-speed routers compared with circuit switching will lead to a migration from IP and ATM packets over SONET to packets over optical fiber directly. That point in time will mark the beginning of the decline of TDM-based circuit-switched systems in today's telephone system.

Further out in the future are purely optical networks. These are already seen today in experimental LANs and access networks. However, optical packet switching is still some ways away.

We begin with a study of the key components of an optical network technology starting with an optical link in section 11.1 and WDM in section 11.2. We describe optical cross-connects in section 11.3 and discuss some of the routing and path-selection problems that this technology poses.

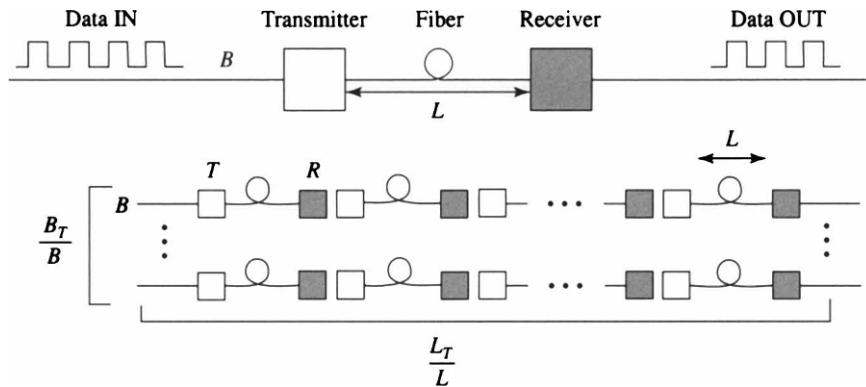
Section 11.4 is devoted to optical LANs, both single-hop and multihop. We will see how questions of performance in these networks are similar to those we studied in Chapters 9 and those that we will address in Chapter 12.

Section 11.5 considers MAN and WAN WDM networks and questions of routing for these networks. We conclude with a discussion of future optical networks.

## 11.1

## OPTICAL LINKS

We view an optical link as consisting of a transmitter, a fiber, and a receiver, as in the top of Figure 11.1. The input bit stream is represented by an electrical input signal (Data IN). The transmitter converts this input signal into an optical signal using a particular modulation scheme. The most important scheme for digital transmission is *on-off keying*, or OOK. In OOK a laser is turned on for a 1 bit and turned off for a 0 bit. Thus a bit stream of 1s and 0s is converted into a sequence of light and dark (no light) pulses. The modulated optical signal propagates over the fiber and reaches the receiver, where it is demodulated into an electric signal (Data OUT) from which the input bit stream is recovered, possibly with error.



11.1

FIGURE

A link of rate  $B_T$  and length  $L_T$  can be built by a series-parallel connection of  $B_T/B \times L_T/L$  links each of rate  $B$  and length  $L$ .

As a communication system, a link is characterized by a pair of numbers  $(B, L)$ . Here  $B$  bps is the bit rate and  $L$  km is the maximum distance of the fiber for which the error rate is below a specified amount. For optical links, this bit error rate or BER is on the order of  $10^{-12}$ . We may transmit  $B$  bps over a distance of  $2L$  km with two  $(B, L)$  links in series: at the receiver of the first link, the original bit stream is regenerated (with some error) and used to modulate the transmitter of the second link. (If each link's BER is  $10^{-12}$ , the series connection BER is of the same order.)

Suppose we want to build a communication system that can transmit  $B_T$  bps over a distance of  $L_T$  km using  $(B, L)$  optical links. We can achieve our aim with  $B_T/B$  parallel systems, each system consisting of  $L_T/L$  links in series, as shown in the bottom of the figure. Thus, we need  $(B_T \times L_T)/(B \times L)$  optical links. Hence, as a communication system, the value of a link that can transmit  $B$  bps over  $L$  km is proportional to the bit rate-distance product  $B \times L$ . If a link has a  $B \times L$  product twice as large as another link, then one should be willing to pay twice as much for it since only half as many are needed.

Link performance is affected by the limitations of its three components: transmitter, fiber, and receiver.

### 11.1.1 Transmitter

The transmitter is a modulated source of light. A laser diode is the light source. (For short distances and relatively low bit rates, a cheaper light-emitting diode or LED may suffice.) The laser, which stands for light amplification by

stimulated emission of radiation, was invented in 1958. We briefly describe the laser mechanism. When an electron decays from one energy state to another, the excess energy is sometimes emitted as a photon of light. This process is called *spontaneous emission*. The wavelength of the emitted photon is inversely proportional to its energy,

$$\lambda = \frac{hc}{W_g} = \frac{1.24}{W_g(eV)} \mu\text{m},$$

where  $h$  is Planck's constant, and  $c$  is the speed of light. The energy  $W_g$  depends on the material of the laser diode. For the gallium arsenide alloys used in laser diodes, the wavelengths  $\lambda$  cover the range 0.8 to 1.7  $\mu\text{m}$  suitable for transmission over optical fibers.

Light amplification is achieved as photons move back and forth between two parallel mirrors, triggering *forced* or *stimulated emission*. Ideal laser light is formed when groups of photons are all in the same phase, or coherent. These two properties of amplification and coherence create the laser's highly directional, pure color beam.

In a semiconductor laser, the intensity of the light is proportional to the injected current. By changing this current according to the signal being transmitted, the intensity of the light is modulated. The receiver demodulates this light and recovers the signal. In digital transmission, the light is turned on or off. In analog transmission, the light is modulated continuously.

The transmitter's limitations are determined by the power  $P_T$  of the light source, its coherence, and its modulation bandwidth, that is, the maximum rate at which the light source can be turned off or on. Laser diodes have an output power of 10 mW and a modulation bandwidth of 3 GHz.

### 11.1.2 Receiver

Modulated light from the transmitter is launched into the fiber. At the distant end of the fiber the receiver converts the optical signal into an electrical signal and demodulates it to recover the modulating signal—the input data at the transmitter.

To determine whether a 1 or 0 is transmitted during a specific bit time requires several operations: photo detection, amplification, filtering, and decision. Photo detection is done by a photodiode, which converts the received optical signal into electric photocurrent. The amplifier converts the photocurrent into a voltage signal at a usable level. The low-pass filter reduces the noise introduced by the amplifier by cutting off frequencies beyond the bandwidth of

the input data signal. The decision circuitry includes an equalizer to restore the data pulse shape and a timing extractor, and it compares the processed signal with a threshold to decide whether a 1 or 0 bit is received.

The voltage signal on which this decision is based is corrupted by three noise sources: the photodetector shot noise, the photodetector dark current, and the amplifier thermal noise.

The photocurrent is not a deterministic process, but a *shot noise* process. It is the sum of a sequence of impulses that coincide with the random arrival times of the photons that constitute the optical signal. (The arrival times have a Poisson distribution.)

The *dark current* is the photocurrent produced even when no external light is impinging on the photodiode. Dark current is caused by the spontaneous thermal excitation of electrons in the photodiode. Typical values of dark current range between 1 and 5 nA (nanoamps).

The *thermal noise* is a white noise process produced by the amplifier. Its power is proportional to the bandwidth of the low-pass filter and hence to the bit rate  $B$ .

The three noise sources are independent, and so their effect is additive:

$$\langle i^2 \rangle_{total} = \langle i^2 \rangle_{shot} + \langle i^2 \rangle_{dark} + \langle i^2 \rangle_{thermal},$$

where  $\langle i^2 \rangle_{total}$  is the variance of the total noise, and the terms on the right are the variances of the three individual noise components. In practice, the thermal noise dominates the other two noise sources.

Because of the noise, the receiver makes errors in detecting the signal. Errors are measured by the bit error rate (BER), which is a function of the signal-to-noise ratio

$$SNR = \frac{\text{Average signal energy per bit}}{\text{Receiver noise power}}.$$

Receiver performance is measured by its *sensitivity*. By convention, this is the minimum received optical power  $P_R$  needed to achieve a BER of  $10^{-9}$ , at a specified bit rate  $B$ . In general, the larger the SNR, the smaller the BER. As a rule of thumb, in order to achieve a BER of  $10^{-9}$ , we need  $SNR \geq 6$ .

The signal power is given by the average photocurrent,  $I_{ph}$ , which is proportional to the received power,  $I_{ph} = R \times P_R$ . Here  $R$  is the responsivity of the photodetector. Thus the average energy per bit is

$$I_{ph} \times T = \frac{I_{ph}}{B} = \frac{R \times P}{B},$$

where  $T = 1/B$  is the bit time. For example, if  $B = 100$  Mbps, then  $T = 10 \times 10^{-9} = 10$  ns (nanoseconds). We can immediately see that SNR is proportional to  $P_R$  and inversely proportional to  $B$  (or  $B^2$  if we take noise power proportional to  $B$ ).

For gallium arsenide photodiodes,  $R = 1$  amp/W for wavelengths  $\lambda \sim 0.8 - 1.5$   $\mu\text{m}$ . For example, if  $P_R = 1 \mu\text{W}$  ( $-30$  dBm), then the average photocurrent is about  $1 \mu\text{A}$  (microamp).

In summary, the larger the bit rate  $B$ , the greater the received power  $P_R$  needed to maintain a specified BER. The received power is proportional to the transmitted power and the characteristics of the fiber.

### 11.1.3 Fiber

As the optical signal propagates over the fiber, it gets distorted due to attenuation and dispersion. Attenuation is the reduction in power in the optical signal, and dispersion is the spreading of a pulse of light. At any given bit rate, the distortion, and hence the error rate, increases with the length of the fiber.

#### Attenuation

The attenuation of a fiber is expressed in decibels per kilometer (dB/km). To explain why these units are appropriate, we first show that attenuation is exponential in the fiber length. Consider an optical fiber propagating a beam of light. Suppose the power of the beam launched into the fiber is  $P_T$ . As the beam travels along the fiber, some of its power is dissipated. Suppose that after traveling  $l$  km of fiber, the power in the beam is  $P(l)$ .  $P(l)$  is proportional to  $P_T$ . We denote by  $a(l)$  the attenuation factor, that is,  $P(l) = a(l)P_T$ . The power in the beam after  $l_1 + l_2$  is  $P(l_1 + l_2)$ , which may be expressed in different ways,

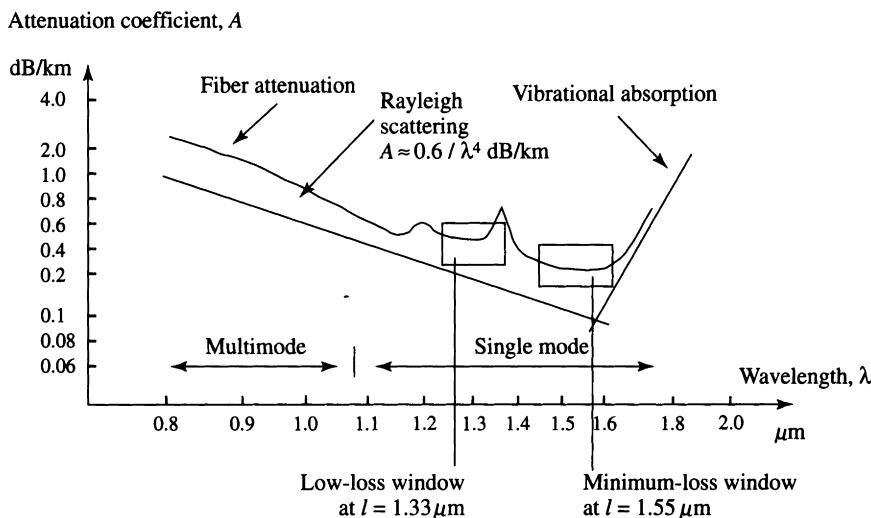
$$P(l_1 + l_2) = a(l_1 + l_2)P_T = a(l_1)P(l_2) = a(l_1)a(l_2)P_T.$$

The first equality follows directly from the definition of  $a(l)$ . The second expression is obtained by writing  $P(l_1 + l_2)$  as the power  $P(l_2)$  attenuated by  $l_1$  km of fiber. We conclude that

$$a(l_1 + l_2) = a(l_1) \times a(l_2),$$

from which it follows that  $a(l)$  must be of the form

$$a(l) = e^{-\alpha l}, \quad l \geq 0.$$



**11.2**  
**FIGURE**

Attenuation in all-glass fiber is measured in dB/km. There are two low-loss windows near  $1.3$  and  $1.55 \mu\text{m}$ .

Since  $\alpha(l) < 1$ , we must have  $\alpha > 0$ . By modifying the expression for  $\alpha$ , the function  $\alpha(l)$  can be rewritten as

$$\alpha(l) = 10^{-\frac{\alpha l}{10}}.$$

The attenuation after  $L$  km is such that

$$10 \log \frac{P_T}{P(L)} = A \times L,$$

so that the attenuation in decibels is equal to  $A$  multiplied by the distance  $L$  in km. Thus,  $A$  is the attenuation of the fiber in decibels per kilometer.

The attenuation coefficient  $A$  of the fiber depends on the fiber material and also on the wavelength  $\lambda$  of the light. Figure 11.2 shows  $A$  for an all-glass fiber as a function of  $\lambda$ , measured in  $\mu\text{m}$  or microns.

The figure indicates two different physical causes of attenuation, Rayleigh scattering and vibrational absorption. There are two "windows" of wavelengths where the attenuation is at a minimum. One of these windows is at  $1.33 \mu\text{m}$ , and its attenuation is  $0.4$  dB/km. The other window is at  $1.55 \mu\text{m}$ , and its attenuation is  $0.25$  dB/km.

The width of each of these windows translates into an enormous bandwidth. For instance, the window around  $1.55 \mu\text{m}$  has a width of  $200$  nm

(nanometer) ( $1 \text{ nm} = 10^{-9} \text{ m}$ ). The range of frequencies of light carried in this window goes from  $c/(\lambda + 200) \text{ nm}$  to  $c/\lambda$ , where  $\lambda \approx 1.45 \mu\text{m}$ , that is, from  $(3 \times 10^8)/(1.65 \times 10^{-6}) = 1.818 \times 10^{14} \text{ Hz}$  to  $(3 \times 10^8)/(1.45 \times 10^{-6}) = 2.068 \times 10^{14} \text{ Hz}$ . Therefore, this window covers a range of frequencies of about  $25 \times 10^{12} \text{ Hz}$  or 25,000 GHz. For all practical purposes the bandwidth of an optical fiber is unlimited.

To utilize this bandwidth, however, requires modulating the laser transmitter at very high speeds. Today's electronics limit the speed to 2.5 Gbps. The limit is likely to increase to 10 Gbps. A much better approach to utilizing the bandwidth is offered by wave-division multiplexing (WDM), discussed later.

We can determine the maximum usable length of an optical fiber from its attenuation coefficient  $A$  if we know the transmitted power  $P_T$  and the receiver sensitivity  $P_R$ . To determine that maximum length, we use the formula expressing the received power  $P(L)$  after  $L \text{ km}$  in which we set  $P(L) = P_R$ , and we solve for  $L$ . This gives  $L = \frac{10}{A} \log_{10} \frac{P_T}{P_R}$ .

It is convenient to express  $P_T$  and  $P_R$  in dBm. By definition, a power  $p$  in watts is equal to  $P(\text{dBm})$ , where

$$P(\text{dBm}) := 10 \log_{10} \frac{p}{1 \text{ mW}}.$$

With this definition, we can rewrite the formula for the maximum usable length  $L$  as

$$L = \frac{1}{A} \{P_T(\text{dBm}) - P_R(\text{dBm})\}. \quad (11.1)$$

We illustrate the use of formula (11.1). Suppose  $P_R = -45 \text{ dBm}$  (about  $3 \times 10^{-8} \text{ W}$ ) at a rate of 1 Gbps and a BER of  $10^{-12}$ . Suppose next that the attenuation is  $A = 0.2 \text{ dB/km}$ . Finally, suppose the transmitter power  $P_T$  is 1 mW (0 dBm). Then the maximum fiber length is

$$L = \frac{1}{0.2} \{0 - (-45)\} = 225 \text{ km},$$

so that the  $B \times L$  product of this link is 225 Gbps·km.

Another way to use the formula (11.1) is to express the power loss as

$$P_T(\text{dBm}) - P_R(\text{dBm}) = A(\text{dB/km}) \times L(\text{km}).$$

Besides attenuation in the fiber, the main causes of power loss between transmitter and receiver are the coupler between the source and the fiber, the splices between sections of fiber, and the coupler between the fiber and the receiver. Thus, if the light between the transmitter and the receiver goes through two couplers,  $N$  splices, and  $L \text{ km}$  of fiber, then the power loss is given by

$$A(\text{dB/km}) \times L(\text{km}) + 2 \times C(\text{dB}) + N \times S(\text{dB}),$$

where  $C$  is the power loss at a coupler (in dB) and  $S$  is the power loss at a splice. The *power budget* analysis of the communication link is the comparison of this power loss with the total acceptable loss  $P_T(\text{dBm}) - P_R(\text{dBm})$ .

The formula (11.1) for the maximum usable length of a fiber also applies to the maximum usable length of a coaxial transmission line. (The formula also is used to compute the range of a cellular radio base station.) The values for the attenuation coefficient, transmitted power, and the receiver sensitivity for coaxial cable are of course different from those for an optical link. Typically, a microwave transmitter with a bit rate of 100 Mbps can inject a power of 1 W into the coaxial cable, so that  $P_T = 30 \text{ dBm}$ . This transmitter power is significantly larger than that transmitted in an optical fiber. A microwave receiver can be made very sensitive and requires only  $P_R = -75 \text{ dBm}$  for a BER of  $10^{-12}$ . Thus a microwave receiver requires much less signal power than an optical receiver to achieve the same BER. The attenuation coefficient of a coaxial cable around 100 MHz is about 30 dB/km, much larger than that for optical fiber. Using the formula then shows that the maximum usable length for the coaxial cable is 3.5 km. This results in a  $B \times L$  product of 0.35 Gbps×km, about three orders of magnitude less than the 225 Gbps×km product for the optical fiber of the example. Recalling our earlier discussion, we can conclude that the economic value of the optical link is three orders of magnitude larger than that of the copper link.

This comparison shows that the dominant advantage of fibers over copper is due to their much lower attenuation over a large range of frequencies. Formula (11.1) also explains that reducing the attenuation coefficient by a factor of 10 increases the maximum length by the same factor, whereas increasing the transmitted power or decreasing the power required at the receiver by a factor of 10 has a much smaller impact on the maximum length.

When the received power drops to the value of the receiver sensitivity, the input bit stream is regenerated and used to modulate the optical signal of the next transmitter in series, as shown in Figure 11.1. Thus regeneration serves to amplify the optical signal power, with a gain  $P_T - P_R$ , which is about 30 to 45 dB. Regeneration requires conversion from the optical to the electric domain. The bit rate of an optical signal that can be amplified by regeneration is limited by the maximum bandwidth of electronic amplifiers, which is a few GHz.

Optical amplifiers can increase the power of the optical signal without converting first into an electric signal. Erbium doped fiber amplifiers (EDFA) are now practical so the need for (electronic) signal regeneration is reduced, and the bandwidth-delay product is increased. However, the gain of power amplifiers is not greater than regeneration. The decisive advantage of EDFA

is their enormous bandwidth. They have a passband of 35 nm versus the 200 nm fiber bandwidth, which reduces the overall bandwidth from 25,000 to 5,000 GHz. Wave-division multiplexing is the only modulation scheme that makes use of this bandwidth.

Three generations of optical links have been used to date. We summarize the characteristics of the transmitter, receiver, and fiber used in each generation. The first generation used an AlGaAs (aluminum gallium arsenide) laser or LED as the optical power source providing  $P_T = 1$  mW at a wavelength of 0.85  $\mu\text{m}$ ; multimode fibers (with a core diameter of 50  $\mu\text{m}$  compared with 8  $\mu\text{m}$  for single-mode fibers) with an attenuation coefficient  $A = 2.5$  dB/km; and silicon PIN diodes or avalanche photodiodes (APD) as detectors with a sensitivity of  $\bar{N} = 300$  photons per bit for  $\text{BER} = 10^{-9}$ .

Receiver sensitivity expressed as  $\bar{N}$ , the average number of photons received per bit, can be converted into required receiver power  $P_R$  by the formula

$$P_R = \bar{N}Bh\nu = 2 \times 10^{-7} \times \bar{N} \times \frac{Bb}{\lambda \mu\text{m}} \text{ mW} \approx 7 \times 10^{-14}B \text{ mW.}$$

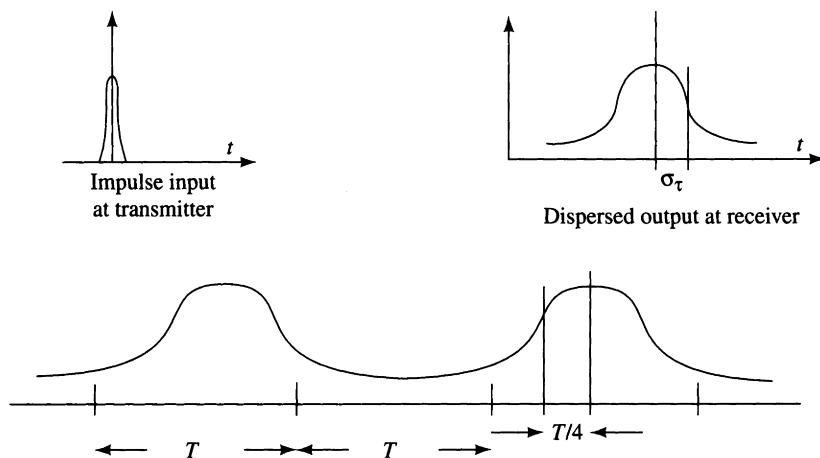
Here  $B$  is the bit rate in bps.  $P_R$  is obtained by multiplying  $(\bar{N} \times B)$ , the average number of photons per second, by the energy ( $h\nu$ ) of each photon at the frequency  $\nu = c/\lambda$ . For instance, if the receiver needs 300 photons per bit, then its sensitivity is equal to  $7 \times 10^{-14}B$  mW, when the transmission rate is  $B$  bps.

The second-generation optical link used lasers with  $P_T = 1$  mW in the low-loss window of 1.3  $\mu\text{m}$ , single-mode fiber with attenuation coefficient  $A = 0.4$  dB/km, and InGaAs (indium gallium arsenide) PIN or APD diodes as detectors with a sensitivity of  $\bar{N} = 1,000$  photons per bit for a BER of  $10^{-9}$ . The third generation uses lasers with  $P_T = 1$  mW in the minimum-loss window at 1.55  $\mu\text{m}$ , single-mode fiber with attenuation coefficient  $A = 0.25$  dB/km, and with a receiver similar to that of the second generation.

Thus the principal advance from one generation to the next is the reduction in attenuation coefficient. First-generation links are used where the distance between transmitter and receiver is short, so that the small distance-bandwidth product is not a limitation. (For very short distances, copper coaxial cable may be sufficient.) Third-generation links are used for long distances.

### *Dispersion*

The top of Figure 11.3 shows that a narrow pulse representing a 1 spreads as it travels down the fiber, with a spread designated by  $\sigma_\tau$ . To understand how dispersion limits the bandwidth-distance product, suppose the transmitter turns the source of light on for  $T$  seconds to represent a 1 and turns it off for



**FIGURE**  
11.3

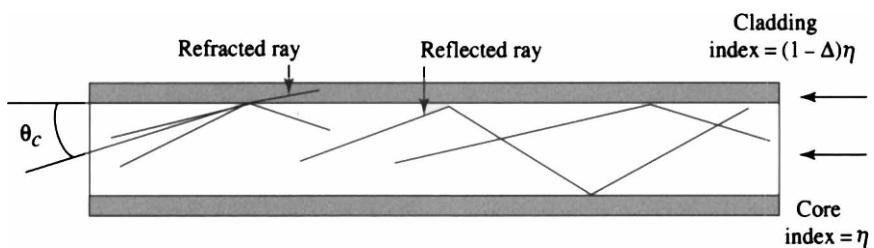
A narrow pulse representing a 1 spreads as it travels down the fiber. The dispersion should be less than one-quarter of the bit time,  $T/4$ , to prevent errors.

$T$  seconds to represent a 0. ( $T = 1/B$  is the bit time, where  $B$  is the bit rate in bps.) Thus, the transmitter represents the input bit string by a succession of light (and dark) pulses. To recover the bits, the receiver must distinguish the periods when the light is on from those when it is off. Consequently, as the pulses representing 1s spread, they overlap epochs that represent 0s; the dispersion will not confuse the receiver if the pulse spread is less than  $T/4$ . In that case, as seen in the bottom part of Figure 11.3, the receiver will see a 0 between two 1s as a small period when the light is off. However, if the pulse spread  $\sigma_\tau$  is larger than  $T/4$ , then the receiver may not be able to distinguish the 0s and the 1s.

The pulse spread is equal to  $\alpha L$ , where  $\alpha$  is a constant that depends on the fiber. Consequently, if the pulse spread is to be less than  $1/4B$ , then  $\alpha L \leq 1/4B$ , that is,

$$B \times L \leq \frac{1}{4\alpha}.$$

Thus, dispersion limits the bandwidth-distance product. This dispersion limit, together with the attenuation limit, determines the maximum usable length of the fiber at a given bit rate. The dispersion limit depends on the fiber (through the coefficient  $\alpha$ ). We will explain the physical cause of this limit for the following types of fibers: step index, graded index, and single mode.



11.4

FIGURE

In a step-index fiber, light rays propagating at angles less than  $\theta_c$  are reflected into the fiber.

Figure 11.4 shows a step-index fiber. It consists of a cylindrical core made of a material with refractive index  $\eta$  surrounded by a cladding made of a material with refractive index  $(1 - \Delta)\eta$ . For all-glass fiber,  $\eta \approx 1.46$  and  $\Delta \approx 0.01$ . The speed of light in a material with refractive index  $\eta$  is equal to  $c/\eta$ , where  $c = 3 \times 10^5$  km/s is the speed of light in a vacuum. Thus, the speed of light in glass is about  $2 \times 10^5$  km/s.

For the step-index fiber, the pulse spread can be computed to be

$$2\sigma_\tau = \frac{L\eta}{c} \left[ \frac{1}{\cos \theta_c} - 1 \right] \approx \frac{L\eta}{c} \left[ \frac{1}{2} \theta_c^2 \right] \approx \frac{L\eta}{c} \Delta.$$

Since the pulse spread should be less than  $1/4B$ , dispersion places the limit

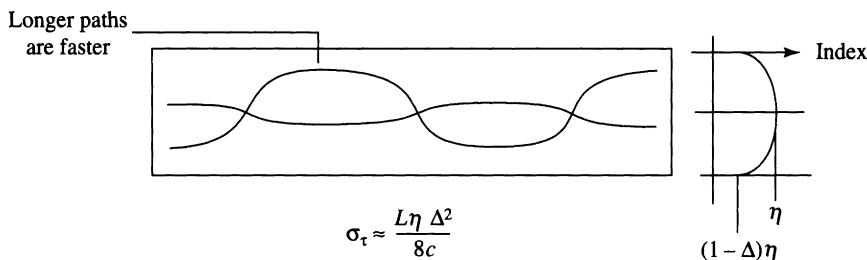
$$B \times L < \frac{c}{2\eta\Delta} = \frac{3 \times 10^5}{2 \times 1.46 \times 0.01} = 10 \text{ Mb} \times \text{km}.$$

This stringent limit, which is worse even than that of coaxial cable, led researchers to design a different type of fiber with a much higher limit. These fibers have a graded-index (GRIN) profile illustrated in Figure 11.5. In such a fiber, the refractive index decreases continuously away from the fiber center, as shown by the parabolic profile on the right in the figure. The pulse spread for a GRIN fiber can be computed to be

$$\sigma_\tau = \frac{L\eta}{8c} \Delta^2.$$

In order to keep  $\sigma_\tau < 1/4B$ , we must then have

$$B \times L < \frac{2c}{\eta\Delta^2} \approx 4 \text{ Gbps} \times \text{km},$$



11.5

FIGURE

In a graded-index (GRIN) fiber, modes that have longer paths travel faster, resulting in lower dispersion than in step-index fibers.

which is two orders of magnitude better than the  $10 \text{ Mbps} \times \text{km}$  limit for step-index fiber.

Step-index and graded-index fibers are called *multimode* because light travels in several different modes in these fibers. The resulting dispersion is called *modal dispersion*. When the core diameter of a step-index fiber is less than  $8 \mu\text{m}$ , Maxwell's equations imply that only one mode can propagate through the fiber. Such fibers are called *single-mode* fibers. There is no modal dispersion in a single-mode fiber. However, there is *material dispersion* due to the fact that light from the laser is composed of different wavelengths, which travel at different speeds.

Material dispersion is computed as follows. Let  $D(\lambda)$  be the linear material dispersion of the fiber, expressed in  $\text{ps}/\text{km}\cdot\text{nm}$ .  $D(\lambda)$  is the difference in travel times in picoseconds ( $1 \text{ ps} = 10^{-12} \text{ s}$ ) for light rays with wavelengths that differ by 1 nm, per km of fiber. Suppose the spectrum of the laser source has a width of  $\sigma_\lambda$ , centered at  $\lambda_0$ . Then the pulse spread

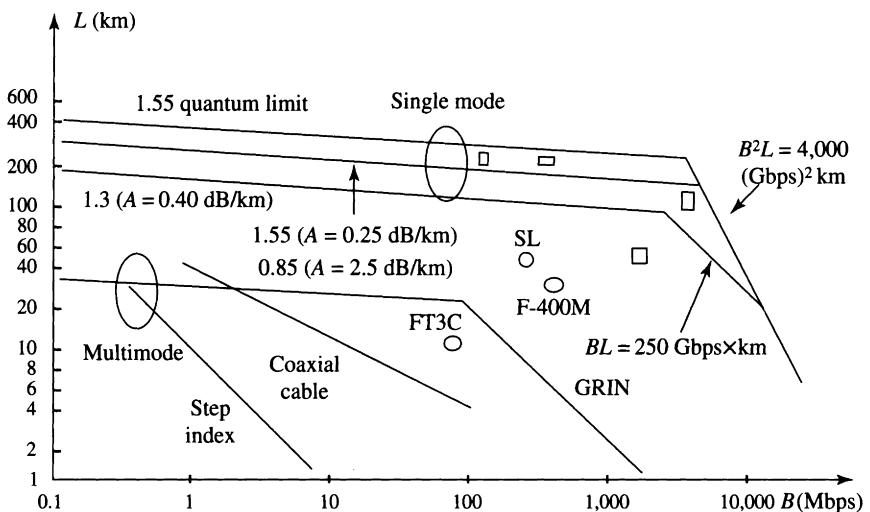
$$\sigma_\tau = L \times D(\lambda_0) \times \sigma_\lambda.$$

In order to achieve  $\sigma_\tau < 1/4B$ , we must have

$$B \times L < \frac{1}{4D(\lambda_0) \times \sigma_\lambda} \approx \frac{1}{4 \times 1(\text{ps}/\text{km}\cdot\text{nm}) \times 1(\text{nm})} = 250 \text{ Gbps} \times \text{km}.$$

The material dispersion limit of  $250 \text{ Gbps} \times \text{km}$  assumes some typical values for  $\lambda_0 = 1.33 \mu\text{m}$ .

Figure 11.6 summarizes our discussion of optical fibers. It shows the maximum repeaterless distance for fibers as a function of the bit rate, without the use of optical amplifiers. The maximum distance is shown for three different types of fiber: step index, graded index, and single mode. The limits on the



11.6  
FIGURE

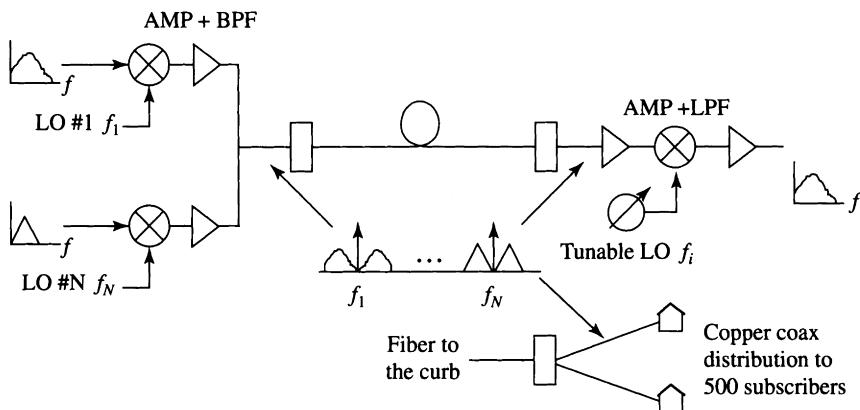
The maximum repeaterless distance as a function of bit rate, for three generations of optical fiber.

distance are determined by the attenuation limit and by the dispersion limit. (In practice, for single-mode fibers the attenuation limit is more important; for multimode fibers the dispersion limit is more important.) Also shown is the theoretical maximum distance assuming a typical transmitted power and the minimum theoretical receiver sensitivity for on-off keying. This minimum sensitivity is called the *quantum limit for OOK*. In addition, the figure shows some values (denoted by small circles or boxes) achieved by experimental or commercial systems.

### 11.1.4 Subcarrier Multiplexing

In OOK the laser is directly modulated by the data stream. Subcarrier multiplexing or SCM, illustrated in Figure 11.7, is a nondirect modulation scheme. SCM is very practical, similar to current radio or TV broadcast, the chief difference being that the transmission medium is optical fiber instead of free space.

In the system envisaged in the figure,  $N$  analog or digital baseband signals modulate different local microwave oscillators at different RF subcarrier frequencies,  $f_1, \dots, f_N$ . The electrical signal obtained by adding the modulated subcarriers now modulates a single laser. (The word *subcarrier* is used to distinguish the local oscillators from the lightwave “carrier”) At the receiver, direct



11.7

**FIGURE**

In subcarrier multiplexing, several signals modulate different RF (radio frequency) subcarriers. The sum of those subcarriers modulates one laser. At the receiver, a signal is recovered by mixing with the appropriate subcarrier. Fiber to the curb uses subcarrier multiplexing.

detection is followed by “downconverting” to IF (intermediate frequency) or to baseband.

This scheme can be used to combine the separate distribution systems of cable TV, telephone, and data networks into a single fiber. Suppose the total bandwidth available to modulate the lightwave carrier is 500 MHz. A TV signal, after digitizing and compression, may require a bit rate of 2 Mbps. Thus 250 TV channels would require 500 Mbps or occupy a bandwidth of 250 MHz. This would leave about 250 MHz for accommodating voice, data, and other services. Cable TV companies are upgrading their residential copper distribution plant by fiber with subcarrier multiplexing. The plan, called *fiber to the curb*, is illustrated in the bottom right of the figure. The optical fiber is brought to a residential neighborhood (curb), demodulated, and the electrical signal is amplified and distributed via short copper coaxial cable to 500 subscribers. Each subscriber must add equipment (TV set-top box) that will demodulate the subcarriers. Other subscriber receiving equipment, such as TV sets and phones, remains unchanged. (See the discussion on cable TV in section 5.7.)

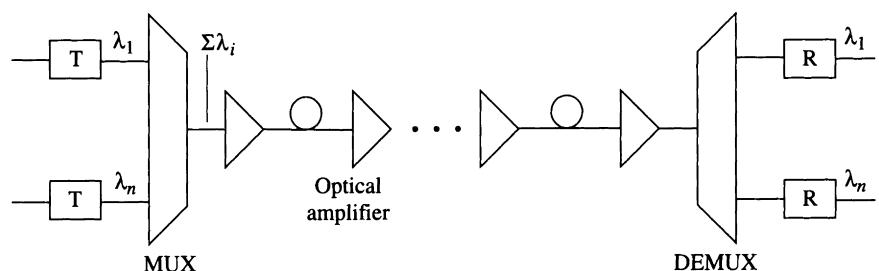
The use of SCM for cable TV distribution is limited by two power factors. The distribution system forms a tree terminating at  $n$  leaves at the curb. Thus, neglecting losses, the power received at each leaf is  $1/n$ th the power inserted at the head end. Second, the sum of the powers of the  $N$  subcarriers has to be limited to the maximum power of a single subcarrier, to reduce deleterious intermodulation.

**11.2****WDM SYSTEMS**

The direct and nondirect modulation schemes considered above modulate a single lightwave carrier. Since the electronic modulation bandwidth of a laser diode is around 3 GHz, these schemes use only a tiny fraction of the 200-nm low-loss windows (25,000-GHz bandwidth) centered at  $1.55 \mu\text{m}$ . Wave-division multiplexing or WDM makes much better use of this bandwidth. WDM divides the window into  $n$  channels centered at different wavelengths or light "colors,"  $\lambda_1, \dots, \lambda_n$ . Light of each wavelength is generated by a separate laser and modulated independently. The  $N$  modulated lightwaves are combined together and transported over the same fiber. At the receiver, a filter selects the desired channel or wavelength, the lightwave signal is demodulated, and the modulating signal is recovered.

Commercial wave-division multiplexers (WDM) combining up to 16 wavelengths were introduced in 1996, 40-channel systems were available in 1998, and 80- and 128-channel multiplexers are expected in 2000. Since WDM equipment can be used with existing fibers, the capacity of those links can immediately be increased from 2.5 to 100 Gbps.

Figure 11.8 depicts a WDM link. The transmit portion comprises  $n$  laser transmitters ( $T$ ), one for each of  $n$  wavelengths,  $\lambda_i$ . The  $n$  modulated lightwaves are combined (multiplexed) by a passive coupler, amplified, and launched into the fiber. The fiber comprises several spans, each terminated by an optical amplifier. The amplifier compensates for the loss in signal strength over one span and extends the length of WDM links without conversion to the electrical domain. The bandwidth of optical amplifiers today is limited to about 5,000 GHz. The

**11.8****FIGURE**

A WDM link consists of a transmitter with a multiplexer (MUX), a fiber of many spans terminating at an optical amplifier, and a receiver with a demultiplexer (DEMUX).

number of spans that can form a single link, before signal regeneration is required, is limited by the distortion introduced by the fiber nonlinearities and the amplifier noise.

At the end of the link the received light signal is amplified and demultiplexed. This is done by passively splitting the signal into  $n$  copies (each with  $1/n^{\text{th}}$  power) and then passing the  $i^{\text{th}}$  copy into a filter tuned to the  $i^{\text{th}}$  wavelength. The filter output is then processed to recover the  $i^{\text{th}}$  data signal.

The number of wavelengths that can be multiplexed is limited by the stability and wavelength-resolving ability of tunable transmitters and receivers. Today, laser tunability permits typical channel spacings of 0.5 to 1.0 nm (60 to 120 GHz), and tunable receivers can resolve 100 channels.

WDM offers *protocol transparency* since each wavelength is modulated independently. So if WDM and optical amplification are used over a link between a pair of end nodes, the signal carried by each wavelength can support its own signal format and protocol and need not conform to any particular common protocol. Thus one wavelength may carry analog TV signals, another may carry SONET, and a third may carry IP packets.

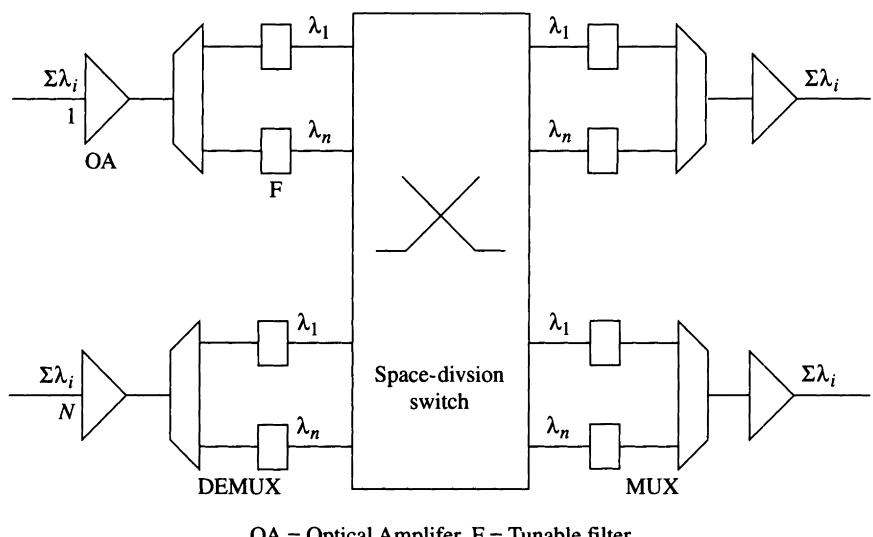
A limited form of transparency is achieved even when there is conversion from optical to electric domain provided that the modulating signal can be recovered from the transition times between 0s and 1s, as in on-off keying and FM. In this approach the WDM signal is demultiplexed, the optical signal in each wavelength is converted to electrical, the 0–1 transition times of the electrical signal are recovered, followed by regeneration of the optical signal with increased power and remultiplexing of the different wavelength signals. This is called *bit-level* or *digital transparency*. The conversion is transparent to bit rates, for instance, but the phase, frequency, and analog information contained in the incoming optical signal is lost. Note that in this approach the regenerated optical signal may be at a different wavelength than the incoming signal, that is, the approach permits wavelength conversion.

Protocol transparency is most effectively used in networks that carry heterogeneous traffic. But preserving protocol transparency over a network path with two or more links requires the capability to connect WDM links without conversion to the electrical domain. This is achieved by optical cross-connects.

## 11.3

## OPTICAL CROSS-CONNECTS

Figure 11.9 depicts the functional architecture of a reconfigurable *optical cross-connect* (OXC), also called a *frequency-* or *wavelength-selective switch*. Each of the  $N$  input fibers carries  $n$  WDM channels. After demultiplexing, the  $nN$



**11.9**  
**FIGURE**

Architecture of an optical cross-connect with  $N$  input, output fibers. After the individual channels are demultiplexed and routed through the switch, they are recombined.

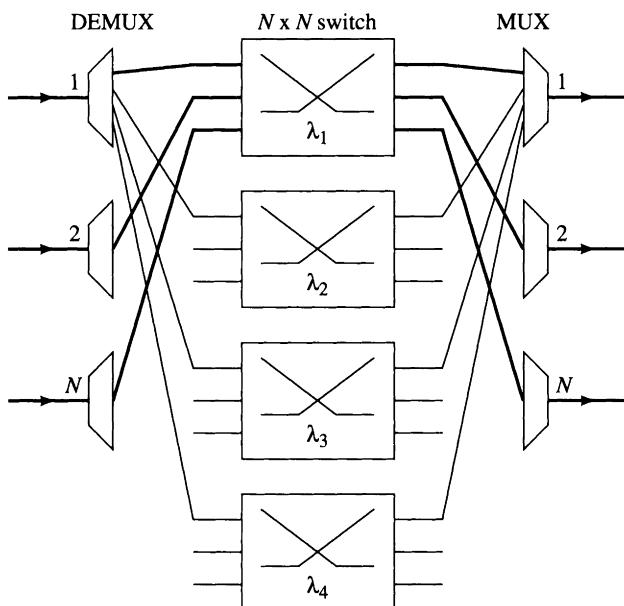
individual channels are switched through a  $nN \times nN$  space-division switch. The switch fabric permutes the  $nN$  channels. The  $nN$  output channels are then remultiplexed into the  $N$  output fibers.

The switch fabric functions like a crossbar discussed in Chapter 12, except that the permutation of the  $nN$  channels must satisfy the restriction that two channels of the same wavelength must be routed to different output fibers. The figure does not show the control mechanisms that configure the switch fabric.

There are variations on the functionality. First, some of the channels may be terminated locally, and local channels may be substituted. The switch may not be reconfigurable. This yields an optical add-drop multiplexer (ADM).

Second, *wavelength conversion* may be possible, so at the output of the switch fabric, a channel with wavelength  $\lambda_i$  may be converted into one with wavelength  $\lambda_j$ . This will overcome the restriction on the permutation.

There are several ways of building a wavelength converter. The most straightforward is to convert the optical signal of one wavelength into an electrical signal and then use the signal to modulate a laser of the appropriate wavelength. At present this is the least expensive approach. But the optical-to-electric conversion destroys protocol transparency.



11.10

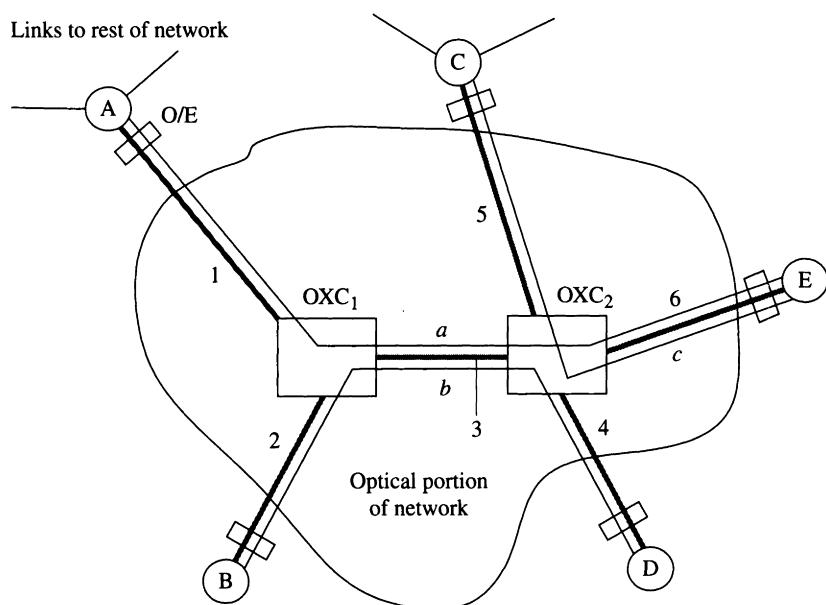
FIGURE

The space-division switch uses a separate switching fabric for each wavelength.

Other promising approaches are based on optical gating and wave-mixing. Gating employs an optical device whose characteristic changes with the intensity of the input optical signal. This change is monitored and used to modulate a continuous wave signal of the desired wavelength. This approach only preserves amplitude and so does not guarantee full protocol transparency. Wave-mixing converters rely on a nonlinear optical response of the medium to the presence of more than one wave. The mechanism preserves phase and frequency as well as amplitude, so it offers full transparency.

The switch fabric itself operates entirely in the optical domain and consists of  $n$  separate fabrics, one for each wavelength, as illustrated in Figure 11.10 for the case  $n = 4$ . The  $N$  channels corresponding to the same wavelength are permuted by the corresponding fabric. The individual fabrics may be built from  $2 \times 2$  switching elements in a modular manner described in section 12.3. Cross-connects today use digital switching fabrics, so that optical-to-electric conversion is first required for each channel. This destroys protocol transparency.

Optical cross-connects, with or without wavelength conversion, permit *wavelength routing*. A virtual *lightpath* can be created that spans several links



11.11

FIGURE

Lightpath *a* spans links 1, 3, 6; lightpath *b* spans links 2, 3, 4; lightpath *c* spans links 5, 6.

joined by cross-connects. A lightpath must carry the same wavelength (in the absence of wavelength conversion). This is called the *wavelength continuity requirement*. The situation is depicted in Figure 11.11, which shows an optical network embedded in a digital network.

The optical network comprises six WDM links, 1, . . . , 6, two cross-connects OXC<sub>1</sub>, OXC<sub>2</sub>, and no wavelength converters. The OXCs are configured in such a way that links 1, 3, and 6 support lightpath *a*. This means that a signal can be transmitted along *a* at the *same* wavelength. Similarly, lightpath *b* is supported by links 2, 3, and 4, and *c* by links 5 and 6. Paths *a* and *b* must be supported by different wavelengths, since they both share link 3. Similarly, *a* and *c* must be supported by different wavelengths since they share link 6.

Lightpaths are the counterpart of SONET paths, described in Chapter 5. SONET paths can be constructed through any contiguous set of links. But the wavelength continuity requirement further restricts the lightpaths that can be formed. We discuss later the resulting wavelength assignment problem.

## 11.4

## OPTICAL LANS

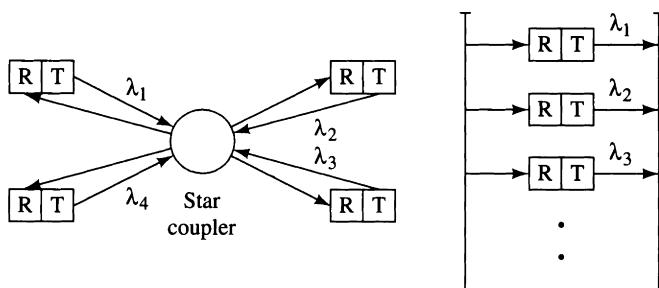
Passive optical  $1:n$  splitters divide a signal into  $n$  parts, each with  $1/n$ th power of the original signal. Passive optical  $n:1$  couplers produce the sum of  $n$  different signals. These passive devices are simple and easy to maintain. We saw their use in cable TV distribution and passive optical networks of section 5.4.

### 11.4.1

### Single-hop LANs

Figure 11.12 illustrates two optical LANs. The star coupler on the left combines the signals from the four transmitters and splits it into four signals sent to each receiver. In the bus arrangement on the right, the signal transmitted by each signal is coupled into the bus. The signal on the bus is split to feed each receiver.

Thus the signals from all the transmitters are broadcast to all the receivers. Each station transmits on one wavelength, but it receives all the wavelengths. Two arrangements are possible. Either each station has a tunable transmitter (laser) that can be tuned to the desired wavelength, or it has a tunable receiver that can select the desired wavelength. The convention is to write FT or TT for a fixed or tunable transmitter respectively, and similarly FR or TR for receivers. In the arrangements shown in the figure, a fixed wavelength is assigned to each station for transmission, so these are FT-FR stations. In the figure, a different wavelength is assigned to each station. This is clearly not necessary. Stations



11.12

FIGURE

Two single-hop LANs. The arrangements assume that the transmitters are fixed and the receivers are tunable.

with different wavelengths may transmit simultaneously without interference, so in theory the total capacity is the sum of the capacities on all the wavelengths.

If station  $i$  wishes to transmit to station  $j$  on wavelength or channel  $\lambda$ , it must make sure that  $j$ 's receiver is tuned to  $\lambda$  and that no other station will transmit on the same wavelength. A protocol is then necessary to coordinate the receivers and transmitters.

Alternatively, the LAN could function like an Ethernet, with no coordination protocol. Collisions can occur, they are sensed, and the transmitter then backs off for a random amount of time before retrying. Suppose the distance between LAN stations is 5 km, the transmission rate is 1 Gbps, and transmitters send 1-Kb packets. The propagation time is then 10  $\mu$ s (2  $\mu$ s per km) and the packet transmission time is 1  $\mu$ s. So the ratio  $a$  of the propagation to the transmission time is 10. As we saw in section 3.2.2, the efficiency of this scheme is  $1/(1 + 5a) = 1/50$ , which is very low. So CSMA techniques are not suitable for optical LANs, unless the transmission times, i.e., the packets, are very large.

If transmission times are small, pretransmission coordination protocols are essential. Two arrangements have been proposed and analyzed. A fixed assignment technique is similar to TDM: a fixed frame duration is selected, divided into slots, and each slot is assigned to a source destination pair  $(i, j)$  of stations. Station  $i$  then gets to transmit to station  $j$  during their slot, in each frame. (This scheme is used in the TPON system of section 5.4.2.) The coordination protocol for fixed assignments is simple. However, a fixed assignment scheme is inefficient if traffic is bursty.

The alternative arrangement is to use a reservation scheme. Time is again divided into frames. To set up a connection from node  $i$  to  $j$ ,  $i$  continuously broadcasts a connection request message, while its receiver is tuned to listen to  $j$ 's acknowledgment. Node  $j$ , if idle, uses its tunable filter to poll across all wavelengths looking for such a request, and locks onto this wavelength if the message is found. It then sends an acknowledgment, which  $i$  is expecting. After  $i$  receives the acknowledgment, a full-duplex path is established, and they can exchange data. The performance of the LAN is affected by the time it takes for a receiver to tune its wavelength, the duration of each connection, and the propagation delay. Rainbow-II is such a network with 32 1-Gbps nodes over a distance of 10 to 20 km.

Many other MAC protocols have been proposed and analyzed. Some protocols need an additional control channel for coordination protocols, and stations may be required to continuously monitor the common control channel.

The analysis of these protocols usually proceeds by building a Markov-chain queuing model of the kind described in section 12.5. We give one example. Time is slotted, data is in packets, with one slot required to transmit one

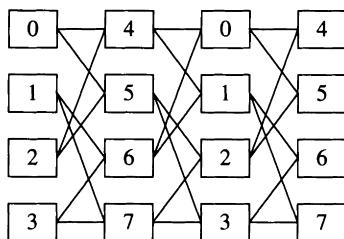
data packet. Packets arrive randomly in each station. A common control channel is used by each station to request permission from a receiver to transmit during the next slot. If permission is received a packet is transmitted, otherwise it is queued. This protocol can be modeled exactly as an input-buffered switch as described in section 12.8. If the queues at each station are FIFO queues, the scheme suffers from head of line (HOL) blocking, and, when the number of stations is large, under uniform traffic, the throughput is 58% as shown in section 12.8.1. If stations can look ahead into the queue, the throughput is increased.

### 11.4.2 Multihop LANs

The LANs of Figure 11.12 are single-hop, since each station is one hop away from the others. Single-hop LANs don't scale easily and require tunable receivers. An alternative arrangement is offered by multihop networks with fixed receiver and transmitter wavelengths.

Figure 11.13 is an example of an 8-node *shuffle network*. Each node transmits and receives two wavelengths. Suppose  $\lambda_1$  is used on the upper link and  $\lambda_2$  on the lower link. (Links are directed from left to right.) If node 0 wants to transmit a packet to node 6, it can send the packet via the route 0-4-1-6 or 0-5-3-6. The links along these paths use different wavelengths, so the arrangement assumes that wavelength conversion is possible at each node. Note that the same wavelength can be used simultaneously on two different hops. Shuffle networks have been extensively analyzed.

We describe one such study for a different realization of a shuffle network. Consider  $N$  stations and a fixed number  $W$  of wavelengths  $\lambda_1, \dots, \lambda_W$ . Each station has one input and one output fiber connected to an OXC like the one



11.13

FIGURE

A multihop shuffle network. Each station transmits and receives on two fixed wavelengths.

shown in Figure 11.10. (Thus the input and output fibers of the OXC with the same number attach to the same station.) Furthermore, each station is able to perform wavelength conversion.

The OXC has  $W$  switching fabrics. Each fabric has a *fixed* configuration. The configuration of the fabric for wavelength  $\lambda_i$  is described as a permutation  $\pi_i$  of the set  $\{1, 2, \dots, N\}$  of stations, so  $\pi_i(m) = n$  means that this fabric connects the output fiber from station  $m$  to the input fiber of station  $n$ .

Each station has an array of  $W$  transmitters and receivers at wavelengths  $\lambda_1, \dots, \lambda_W$ . Data is transferred in fixed size packets. Time is slotted, and the duration of one slot is the time to transmit one packet. Suppose  $\pi$  is the permutation such that station  $i$  wants to set up a connection to station  $\pi(i)$ , for each  $i$ . There is a preassigned schedule of wavelengths  $\lambda_{i(1)}, \dots, \lambda_{i(T)}$  of length  $T$  depending on  $\pi$  such that each station can transmit one new packet of data every  $T$  slots. This works as follows. During slot 1, each station transmits its data at wavelength  $\lambda_{i(1)}$ , and during slots  $2 \leq t \leq T$ , each station recirculates at wavelength  $\lambda_{i(t)}$  the packet it received at wavelength  $\lambda_{i(t-1)}$  on the previous time slot. Thus at the end of the  $T$ th slot, the original data packets have been permuted by

$$\pi_{i(T)} \circ \pi_{i(T-1)} \circ \dots \circ \pi_{i(1)}.$$

If this permutation is equal to the desired permutation  $\pi$ , then each packet can transmit a new data packet in slot  $T + 1$ .

The scheme raises two questions. The first question is whether there is a *fixed* set of permutations,  $\pi(1), \dots, \pi(W)$ , one for each switching fabric, such that any desired permutation  $\pi$  can be realized by an appropriate schedule of these fixed permutations. The answer to this question is simple and elegant: there always exist four permutations that generate the set of all  $N!$  permutations over  $N$  stations. This means that a shuffle network with at least four different wavelengths can realize any permutation, that is, set up any connections among the  $N$  stations.

The second question is to determine the throughput of the network. The number of new data packets per slot transmitted by each station for a permutation that needs  $t$  passes through the network is, of course,  $1/t$ . So the network throughput can be defined in terms of the average number of passes required to realize any permutation. For the case of uniformly distributed traffic, this average number is  $O(N \log N)$  so the average throughput per station is  $O(1/(N \log N))$ .

This shuffle network wastes bandwidth because it uses only one wavelength in each time slot, even though a station is capable of using all  $W$  wave-

lengths simultaneously. More elaborate schemes make use of all wavelengths. Suppose each station can make  $W$  simultaneous connections. Suppose also that as each packet is recirculated, the station can assign it to any wavelength. (Of course, different packets in the same time slot are assigned different wavelengths.) Then the per station achievable throughput is  $1/(3 \log_W WN)$ . For  $W = 4$ ,  $N = 64$ , this gives a per station throughput of  $1/12$ .

Thus, besides being scalable, shuffle networks achieve better bandwidth utilization than single-hop networks. However, they require more elaborate synchronization and control.

## 11.5 OPTICAL PATHS AND NETWORKS

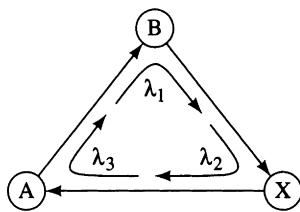
Figure 11.11 illustrated the use of OXCs to create lightpaths traversing several links in a WAN, similar to SONET paths. The network of the figure is an arbitrary mesh. We discuss management and performance of these networks.

### 11.5.1 Static Wavelength Assignment

From the perspective of the external network in Figure 11.11, the optical network is equivalent to a network with three logical links,  $a$ ,  $b$ , and  $c$ . (Logical links are defined in the same way by digital cross-connects as in Figure 5.6.) By configuring the OXCs differently, the optical network can realize different sets of logical links or lightpaths. Depending on the pattern of traffic, one set of lightpaths will be better than another. This suggests the question of finding the best set of lightpaths. This is sometimes called the *wavelength assignment* problem. We study various versions of this problem.

To state matters precisely, we define a *lightpath* to mean a route together with a fixed wavelength assigned to that route. Then a set of lightpaths is feasible if two routes that share the same link are assigned different wavelengths. Thus, for example, in Figure 11.11, lightpaths  $a$  and  $b$  must be assigned different wavelengths because they share link 3, and  $a$  and  $c$  must be assigned different wavelengths because they share link 6.

We describe the physical topology of the optical network as a directed graph  $G = (V, L, W)$ , where  $V = \{1, \dots, N\}$  is the set of vertices or nodes and  $L \subset V \times V$  is the set of directed edges:  $(i, j) \in L$  if and only if there is a physical link or fiber from node  $i$  to  $j$ . Lastly,  $W$  is the number of wavelengths available on each link.



11.14

FIGURE

The NWC number for the three routes is 2, but the SWA number is 3.

Suppose we are given a set of routes. (The routes might be derived from the requirements of the overall traffic pattern.) The *static wavelength assignment* (SWA) problem is to determine an assignment of the given routes to wavelengths such that two different routes are assigned different wavelengths if they share a common link. We adopt a slightly different viewpoint and try to determine the minimum number of wavelengths that are needed to support the given set of routes.

The SWA problem turns out to be difficult: it is NP-complete. There is a weaker version of the problem that is easy. Suppose we dispense with the *wavelength continuity* requirement that a route must be assigned a fixed wavelength. Then the minimum number of wavelengths that one needs is simply the maximum number of routes that traverse a single link. We can find this number easily: for each link we find the number of routes through that link and find the largest number. We call this the non-wavelength continuity (NWC) number of wavelengths.

Clearly the NWC number is smaller than the SWA number. Figure 11.14 shows a network with three links and three routes, each of which traverses two links. Since each link supports two paths, the NWC number is 2. However, with the wavelength continuity requirement, each path must be assigned a distinct wavelength, so the SWA number is 3.

The graph of Figure 11.14 contains loops. A graph without loops is said to be *acyclic*. It is not difficult to see that for acyclic graphs, the NWC number is the same as the SWA number. Also note that if we are free to use wavelength converters, then the NWC and SWA numbers are the same, because we can always change the wavelength of a route on a link if another route through the link has the same wavelength.

The SWA problem for a network with loops is difficult to solve exactly, and so one needs to solve it using heuristics. One good heuristic is the following greedy algorithm. We sort the given set of routes by length. We assign a wavelength to the longest route. We go down the list and assign the same

wavelength to a route provided it does not share a link that supports another route to which the same wavelength has already been assigned. If we cannot assign an existing wavelength to any more routes, we assign a new wavelength to the longest unassigned route. We proceed in this way until all routes have been assigned.

If we apply the greedy algorithm (or any other algorithm for that matter) to the SWA problem, we may end up assigning more wavelengths than are available. In this case, of course, it is not possible to support all the routes. This raises the question of which routes to support. A good approach to formulating this question is to place it in a larger context that includes the traffic between nodes that the network is intended to serve.

We begin again with a physical description of the network as a directed graph  $G = (V, L, W)$ . We are also given a set of possible routes  $R$ . Let  $A_{lr} = 1$  or 0, accordingly as link  $l$  is or is not included in route  $r$ . Let  $z_{wr} = 1$  or 0, accordingly as wavelength  $w$  is or is not assigned to route  $r$ . Each choice of the  $\{z_{wr}\}$  corresponds to a wavelength assignment. If for a route  $r$ ,  $z_{wr} = 0$  for all  $w$ , this means that route  $r$  is inactive. The requirement that active routes through the same link must be assigned different wavelengths is expressed as

$$\sum_{r \in R} A_{lr} z_{wr} \leq 1, \quad \text{for each } l \in L, \quad 1 \leq w \leq W. \quad (11.2)$$

Now suppose we are given a set of source-destination pairs  $P \subset V \times V$ , and let  $B_{pr} = 1$  or 0, accordingly as route  $r$  connects the source destination pair  $p \in P$ . Lastly, let  $x_{pr} = 1$  or 0, accordingly as route  $r$  is or is not dedicated to path  $p$ . We must have the physical restriction

$$x_{pr} \leq B_{pr}, \quad \text{for each } p, r. \quad (11.3)$$

Each choice of the  $x_{pr}$  is a routing assignment, that is, an assignment of source-destination pairs to routes. The routing and wavelength assignments are jointly constrained by

$$\sum_{p \in P} x_{pr} \leq \sum_{w=1}^W z_{wr}, \quad \text{for each } r \in R, \quad (11.4)$$

which says that the number of source-destination pairs assigned to a route  $r$  cannot exceed the number of wavelengths assigned to  $r$ . Inequalities (11.2) through (11.4) define the set of feasible wavelength-routing assignments. We can compare different assignments from this set by a revenue function. Here is one example. The number of paths allocated to the source-destination pair

$p$  is  $\sum_r x_{pr}$ . If  $f_p$  is the revenue generated by a path that serves pair  $p$ , then the revenue-maximizing assignment is given by

$$\max \sum_{p,r} f_p x_{pr}$$

subject to the constraints (11.2) through (11.4). This is a linear integer programming problem. The complexity of the problem grows rapidly with the size of the network, and one needs to resort to heuristic approaches that find good assignments.

This formulation assumes that OXCs are configurable. In practice, the configuration might be changed over the course of a day to match the changing pattern of traffic, using an optimization problem similar to one proposed here. Finally, although the formulation here assumes no wavelength conversion, a straightforward modification of the discussion will lead to a formulation that accommodates wavelength conversion.

### 11.5.2 Dynamic Wavelength Assignment and Blocking

The SWA problem is called “static” because we are free to assign any wavelength to any route. Such a formulation is reasonable when we are designing a WDM network for a target set of routes and the traffic carried by them. In an operational context, we may wish to assign wavelengths to routes dynamically.

Suppose an optical network of the kind depicted in Figure 11.11 offers a service that meets requests for lightpath connections between nodes. Since the lightpath preserves protocol transparency, the service provides a *clear channel* that can be used for any purpose, including data transmission at 2.5 Gbps. A connection, once established, lasts a random amount of time—the connection holding time. Conceptually, this service resembles other circuit-switched connections, like the telephone service. Of course the bandwidth involved is so large that the service wouldn’t be requested by end users, but by hosts such as supercomputers. The service could also be used by network access providers to trade “bulk” connections over the backbone network.

When a connection completes, the link-wavelengths along the lightpath become available for reassignment to meet a new request. The network may be unable to meet a new request because it cannot create a new lightpath from the unassigned link-wavelengths. The request is then *blocked*. The blocking probability suffered by users of this service is then a natural performance measure. Another measure might be network utilization. At first glance, evaluation of

the blocking probability appears identical to the evaluation study carried out in sections 8.2 and 9.2 for circuit-switched telephone networks. There is one difference, however.

Suppose the physical optical network is as depicted in Figure 8.6. If each link supports  $W$  wavelengths, then in terms of the figure each link has  $W$  circuits. Now in the case of the telephone network, a request for a connection can be met if and only if there is a route connecting the source and destination that has one free circuit in each link along the route. In the case of the optical network, a request can be met only if in addition there is a common free wavelength in each link along the route. Of course, if wavelength conversion is possible at every link, then the additional condition disappears, and the discussions in sections 8.2 and 9.2 apply without change.

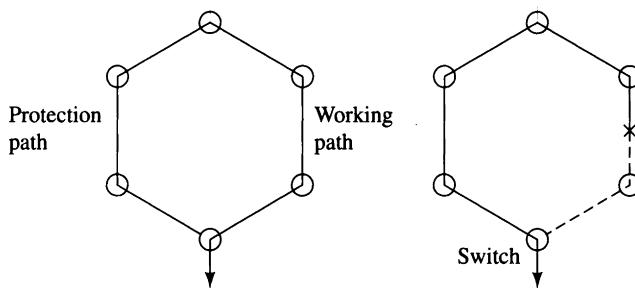
If limited or no wavelength conversion is possible, then the blocking probability clearly must be greater, depending on the network topology, pattern of traffic, and routing algorithm employed. It is possible to carry out the analysis very similarly to that in section 9.2. Closed-form expressions cannot be obtained except for special cases, and you have to resort to numerical methods or simulation. As expected, the larger the number of hops along the routes, the greater the probability of blocking when wavelength conversion is not possible.

Two additional considerations are worth noting. First, even if a new request cannot be met from the available link-wavelengths, it may be possible to meet it if we were to reassign the link-wavelengths of the existing requests. This is the same distinction that is made regarding strict versus rearrangeably nonblocking switches in section 12.3. In practice, such a reassignment is not carried out except under conditions of link or node failure.

Second, in practice it is unlikely that users need the full bandwidth of a lightpath. Usually SONET multiplexers aggregate lower-rate calls onto a wavelength. If these lower-rate calls are indiscriminately multiplexed onto a wavelength, then each wavelength entering or leaving a node (OXC) will need to be converted back to the electrical domain for adding and dropping the lower-rate calls. The multiplexing of these lower-rate calls into a wavelength is called *grooming*. Thus an additional operational decision is to groom calls with an eye toward reducing the number of wavelengths that need to be processed. The usual way of doing this is to identify a few nodes in the networks where such processing occurs, and to "back-haul" the traffic to these nodes.

### 11.5.3 Ring Networks

Although wavelength assignment can be used in WAN mesh networks, its use at present is in MAN ring networks. We consider their application in protection.



**11.15**  
**FIGURE**

Lightpath protected switched ring. Protection switching operates on individual wavelengths.

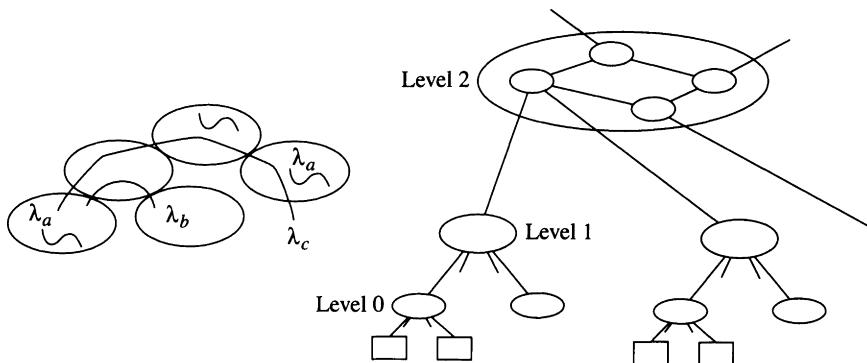
Figure 11.15 depicts an optical ring that protects on a per-channel or wavelength basis, similar to a SONET dual ring. The nodes on the rings are connected with two sets of fibers. Each wavelength that is added to the ring is divided by a splitter and transmitted along the two paths. At the wavelength-dropping switch, both signals are compared and the superior signal is selected, hence no protection signaling is required. Each wavelength is protected independently. Each connection takes up the entire ring capacity for one wavelength, so the maximum number of protected lightpaths is equal to the number of wavelength channels. More elaborate protection schemes are used.

### 11.5.4 Hierarchical Mesh Networks

Very large WDM optical networks that offer switched connections are likely to be organized in a hierarchy to reduce the routing burden. A three-level hierarchy is suggested by Figure 11.16.

A Level 0 subnet is a high-performance LAN like one of the networks considered in section 11.4. Each Level 0 subnet is connected to one Level 1 subnet by a single fiber. The Level 1 subnet might be a metropolitan area network. Each Level 1 subnet is connected to a Level 2 subnet, only one of which is shown in the figure. A Level 0 subnet interconnects several optical terminals (OT).

A subnet at any level can use a preallocated set of wavelengths (labeled  $\lambda_a$ ) for connections that are confined to that subnet. Thus two local connections within two different subnets at any level may use the same wavelength. Connections between two Level 0 subnets connected to the same Level 1 sub-



11.16

FIGURE

A three-level hierarchy for large optical networks. The figure shows partitioning of wavelengths and wavelength reuse.

net can use the wavelengths labeled  $\lambda_b$  for this purpose. Lastly, connections between two Level 0 subnets that must pass through a Level 2 subnet use the wavelengths labeled  $\lambda_c$ . The three sets of wavelengths  $\lambda_a$ ,  $\lambda_b$ , and  $\lambda_c$  are disjoint. An architecture of this kind has been implemented in a 20-wavelength WDM testbed. Three services were offered: a fully transparent, switched lightpath connection between two OTs, a 250- $\mu$ s time-slotted service for lower-bandwidth applications, and a packet-switched low-bandwidth service for signaling.

### 11.5.5 Optical Networks

Current networks use the high-bandwidth optical links as bit ways with a SONET structure. This creates protocol stacks of the form

applications/TCP/IP/ATM/SONET/fiber or applications/ATM/SONET/fiber.

These stacks involve several efficiency losses. IP or layer 3 routing is expensive because of routing table look-ups. This can be overcome by layer 2 routing using tag switching or MPLS. IP packets are grouped into *flows*, encapsulated in tagged packets that are switched in hardware at the link layer and tunneled along ATM virtual paths. The flows must carry sufficient amounts of traffic to make good use of the limited tag address space.

Carrying IP packets over ATM incurs the 10% overhead of ATM headers. Proposals to place IP packets directly in SONET synchronous payload envelopes eliminate this overhead. IP over SONET flows must have speeds that make efficient use of multiple 155-Mbps SONET paths.

A lightpath has speeds of 2.5 Gbps (STS-48) or higher. When flows reach this speed, it will become economical to replace the SONET multiplexing and switching equipment in favor of OXCs and lightpath routing.

Evolution from IP/ATM/SONET/fiber to IP/SONET/fiber to IP/fiber and from ATM/SONET/fiber to ATM/fiber entails a simplification and reduction in equipment costs. The pace of the evolution will be determined by the rate of increase of traffic and the advances in optical networking.

## 11.6

## SUMMARY

Optical communications has revolutionized networking. The first optical links deployed in 1990 soon achieved a bit rate of 2.5 Gbps. By 1998, wave-division multiplexing increased that rate to 100 Gbps, and 1 Tbps (1,000 Gbps) speed will be reached in 2000. At these high speeds, electronic switching becomes a bottleneck, which has spurred advances in optical switching in the form of wavelength-selective switches (OXCs) and wavelength converters. Experience from several testbeds confirms that wide area optical networks can be built to offer permanent or switched-circuit connections at very high bandwidths and with limited or full protocol transparency. Such a service could lead to an active market among backbone network access providers, which could quickly respond to shifts in traffic patterns.

Current technology suggests a limit of 10 Gbps on the achievable data rate of a single channel. Recent laboratory experiments have achieved rates of 100 Gbps with packet switching. These amazing rates are possible using soliton light sources that have pulses of picosecond ( $10^{-12}$  sec) duration. If this technology becomes commercially viable, it will accelerate the creation of pure optical networks.

## 11.7

## NOTES

For a full discussion of the topics covered in section 11.1, see [G93]. A full treatment of subcarrier multiplexing may be found in [M95]. The effects of non-linear distortion and amplifier noise on the scale of a WDM link are analyzed in [TO98]. Advances in tunable lasers and filters are described in [CM98b].

Wavelength converters are reviewed in [So96]. Issues in the design and performance of optical networks using wavelength conversion are reviewed in [RM98].

Optical single-hop LAN protocols are thoroughly discussed in [M92a]. The complexity and performance of protocols for star LANs of Figure 11.12 are analyzed in [CZA93]. The implementation of Rainbow-II is described in [HKR96], and its MAC protocol is analyzed in [JBM96]. The treatment of shuffle networks is adapted from [BCB96].

Recent research in optical networks and descriptions of testbeds are presented in [JSAC96, JSAC98]. The discussion of protection switching is adapted from [M98]. For a comparison of algorithms for span failure restoration see [DGM94].

A treatment with an emphasis on optical networks is [A94]. A complete discussion of wavelength assignment, routing, and performance is given in [RS97]. The SWA problem was first formulated in [CGK92]. It is an NP-complete problem meaning (roughly) that there is no algorithm that can find a solution in the number of steps that is polynomial in the size of the problem. Several different variations of the wavelength and routing assignment problems are discussed in [JSAC96, JSAC98], which also contain several calculations of blocking probability.

For a very informative discussion of optical network technology see [G96]. For a discussion of the issues of wavelength assignment, grooming, and future optical packet networks within the context of ring networks see [M99]. The 20-channel WDM local and MAN testbed is described in [KDD96]. Several ultra-fast TDM networks based on solitons are discussed in [JSAC96].

## 11.8

## PROBLEMS

1. You want to build a 1-Gbps link as long as possible without regeneration. Assume a transmitter power of 1 mW and a receiver sensitivity of  $-30 \text{ dBm}$  for a BER of  $10^{-9}$ . The dispersion at  $1.5 \mu\text{m}$  is  $20 \text{ ps/km.nm}$ , and the attenuation is  $0.25 \text{ dB/km}$ . At  $1.3 \mu\text{m}$ , the dispersion is zero, but the attenuation is  $0.5 \text{ dB/km}$ . What would be the maximum length if you choose (1) a wavelength of  $1.5 \mu\text{m}$  and single-mode, GRIN, or step-index multimode fiber or (2) you choose a wavelength of  $1.3 \mu\text{m}$ ? Assume that the bandwidth of the transmitted signal is about 1 GHz.
2. You want to interconnect a supercomputer to a device at distance 500 m and speed of 1 Gbps. How would you do it? Would you prefer to use coaxial cable or step-index multimode fiber?
3. Power transmitted through an optical fiber attenuates in proportion to distance. Suppose power is transmitted through free space in a small beam of solid angle  $\theta$  for a distance  $L$ . What will be the power incident on a

detector of area  $A$ , ignoring any attenuation due to absorption of light by the atmosphere? What considerations should go into the determination of receiver sensitivity? In outdoor optical communication the ambient sunlight will have a significant impact on receiver sensitivity. How would you account for this in your receiver model?

4. Discuss the following assertions:
  - (a) Dispersion limits the (bit rate)x(length) product.
  - (b) Attenuation limits the length.
  - (c) Dispersion limits the bit rate.
  - (d) If the fiber is single mode, it still has dispersion.
  - (e) WDM increases the bit rate by reducing dispersion.
5. This problem explains how to calculate the BER in on-off keying modulation. The receiver current,  $i(t)$ , is integrated over one bit duration. The integral,  $N$ , is proportional to the energy in the received signal during one bit duration. The value of  $N$  depends on the transmitted signal. When the transmitted signal is 0, the laser source is turned off, the received current is simply noise, and  $N = N_0$ . When the transmitted signal is 1, the laser is turned on, and  $N = N_1$ .  $N_1, N_0$  are Gaussian random variables with the distribution

$$\text{Under 1, } N = N_1 \sim \text{Gauss} \{\bar{N}_1, \sigma_1^2\};$$

$$\text{Under 0, } N = N_0 \sim \text{Gauss} \{\bar{N}_0, \sigma_0^2\}.$$

Thus  $N$  equals  $N_1$  when the transmitted signal is 1 and  $N_0$  when the signal is 0. The receiver compares  $N$  with a threshold and declares that 1 or 0 was transmitted accordingly as

$$N > \frac{1}{2}(\bar{N}_1 + \bar{N}_0) \text{ or } N < \frac{1}{2}(\bar{N}_1 + \bar{N}_0).$$

- (a) Explain why  $\bar{N}_1 > \bar{N}_0$ . Suppose that  $\sigma_i^2$  is proportional to  $\bar{N}_i$ ,  $i = 0, 1$ . Sketch the probability distributions of  $N_0, N_1$ .
- (b) Assume that the transmitted signal is 1 or 0 with probability 0.5. Sketch the probability distribution of  $N$ , and the decision region (i.e., the values of  $N$  for which the receiver declares 1 or 0).
- (c) Show that

$$\text{BER} = \frac{1}{2}[\text{Prob}\{N_1 < \frac{1}{2}(\bar{N}_1 + \bar{N}_0)\} + \text{Prob}\{N_0 > \frac{1}{2}(\bar{N}_1 + \bar{N}_0)\}].$$

6. Recall the definition of the NWC and SWA numbers in section 11.5.1. Show that for an acyclic network the two numbers are the same.

# Switching

**N**eetworks use switching to achieve connectivity among users while sharing communication links. With switches networks can establish higher-capacity communication paths among users with fewer links and at lower per user cost. In this way networks can take advantage of the economies of scale in communication links.

When we view a switch from the outside as a “black box,” it appears as a device with several input and output ports that terminate incoming and outgoing links. An incoming link carries multiplexed bit streams of several users. The switch guides each of these bit streams to the appropriate output port. Networks differ in the methods used to switch and to multiplex signals. The telephone network uses circuit switching and time-division multiplexing. Data networks use packet switching and statistical multiplexing.

In this chapter we explain the operations and the design of switches, concentrating on two types of switches: circuit switches and packet switches. We will see that large switches are composed of small identical modules. This modular design facilitates development, construction, and testing, and it results in expandable systems.

In section 12.1 we review the tasks performed by switches, and we identify useful measures of performance of switches. In sections 12.2 and 12.3 we discuss circuit switching. The concepts introduced in these sections are also useful in the study of packet switching. In section 12.2 we introduce the principles of time-division and space-division switches. In section 12.3 we present the class of Clos networks, and we explain two key results for these networks. One particular switch structure due to Benes is examined in detail. Our discussion

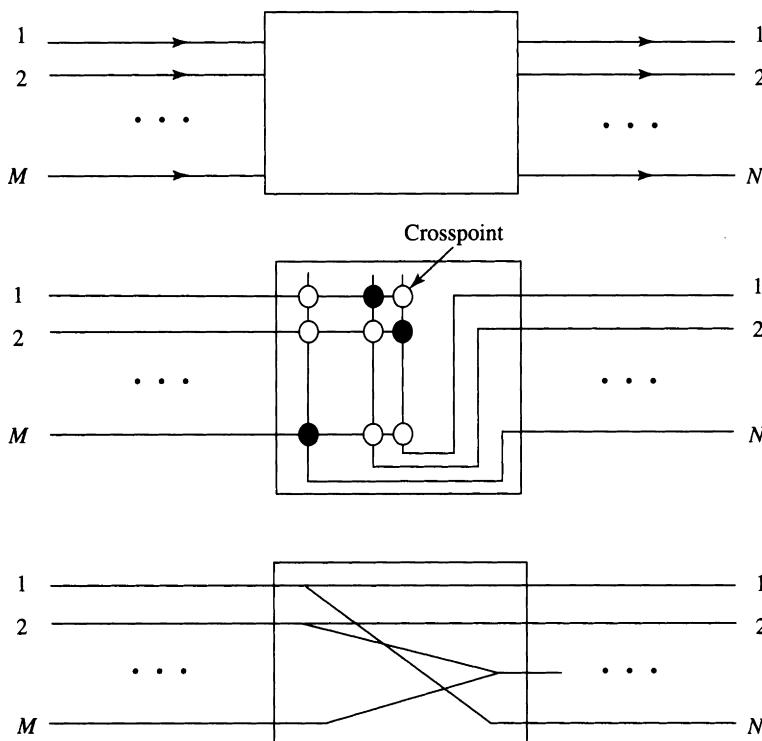
then focuses on fast packet switches. In section 12.4 we present the general operations of these switches, and we introduce four designs: distributed buffer, shared buffer, output buffer, and input buffer. These are studied in sections 12.5 through 12.8. Section 12.9 provides a summary. Note that label switching is discussed in Chapter 4.

## 12.1 SWITCH PERFORMANCE MEASURES

The function of a switch is to provide connectivity among users while sharing resources such as links and buffers. This function is fulfilled by replacing a fully connected network with one in which switches route bit streams along shared links. Since different streams share links, in a circuit-switched network there is the possibility that a call cannot be placed when there is insufficient free capacity to set up a new connection. In a packet-switched network, packets may have to be stored (queued) in a switch before they can be transmitted. Thus, the disadvantage of sharing network resources is the blocking of calls and the queuing of packets. The advantage of sharing is of course a much lower per user cost.

We first regard a switch as a “black box.” The switch now appears as a device with  $M$  input links and  $N$  output links, as in the top of Figure 12.1. (We will also say that the switch has  $M$  input ports and  $N$  output ports. A link is connected to a port.) The function of the switch is to connect input links to output links so that bits or packets that arrive on one link leave on another designated link. Several performance measures can be used to compare switches: connectivity, delay, setup time, throughput, and complexity. We introduce those measures now.

Connectivity is measured by the set of pairs of input and output links that can be simultaneously connected through the switch. The larger this set, the more versatile the switch. As it processes incoming bit streams in order to route them to the proper output ports, the switch introduces delays, and this delay is another measure of performance. For circuit switches, another component of delay is the time needed to set up a circuit. For packet switches, the switch also introduces queuing delay. The throughput of a switch is measured in terms of the number of ports and the speed of the individual input links. Finally, measures to estimate the complexity of a switch include the number of crosspoints (see next page), the buffer size, and the speed of bit streams inside



12.1

FIGURE

Switch as a “black box” (top); a crossbar switch (middle); and a multipoint switch (bottom).

the switch. Switch designs that require higher speeds will need more expensive electronic components.

We now explore one measure of complexity in which a circuit switch is viewed as an organization of crosspoints. Each crosspoint is attached to two links inside the switch. A crosspoint is either closed or open. When it is closed, the crosspoint connects the two links attached to it. When it is open, the two links are not connected. The simplest organization is the *crossbar* switch in which every pair of input and output links is connected by a crosspoint, as in the middle of Figure 12.1. In the figure, the input-output pairs (1, 2), (2, 1), and (M, N) are connected. (A dark crosspoint means it is closed.) More complex organizations are possible where an input link is attached to an internal switch link that is in turn attached to an output link. Multistage switches can be built in this way. Any circuit switch can be viewed as such an arrangement of links

connected by crosspoints, since any routing decision can be decomposed as a succession of binary decisions.

When we regard a switch as an organization of crosspoints, an obvious measure of its complexity is the number of crosspoints. For example, an  $M$  by  $N$  crossbar has  $M \times N$  crosspoints. We will see that there are switches with many fewer crosspoints that can implement almost the same connections as the crossbar.

We define a *switch configuration* as the set of input-output pairs that are simultaneously connected by the switch. Thus, a switch configuration is a subset of the product  $\{1, \dots, M\} \times \{1, \dots, N\}$ , that is, of the set of all input-output pairs. That subset specifies the input-output pairs that are connected. The switch configuration changes when a new connection is made or when an old connection is released. One measure of switch performance is the number of different configurations it can have.

If the switch has  $X$  crosspoints, then it can have at most  $2^X$  different configurations: each crosspoint can be in any one of two states (open or closed) so that the  $X$  crosspoints admit  $2^X$  different states. Different states of the crosspoints may or may not result in different switch configurations, but in any case different configurations must correspond to different states of the crosspoints. Thus, the logarithm (in base 2) of the number of configurations is a lower bound for the number of crosspoints required to build the switch. We state this as a proposition.

**Proposition 12.1.1** If a switch is a collection of crosspoints, then

$$\text{number of crosspoints} \geq \log_2 (\text{number of different configurations}).$$

We consider two important examples of switch complexity. The first example is the  $N \times N$  point-to-point switch. By definition, a point-to-point switch can connect any input link to any output link so long as two different input links are not connected to the same output link and vice versa. Consequently, the configurations of a point-to-point switch are identified by the permutations of the numbers  $\{1, \dots, N\}$ . For instance, the permutation  $231 \dots 7$  specifies that input link 1 is connected to output link 2, input link 2 to output link 3, input link 3 to output link 1, and so on. Different permutations correspond to different configurations and vice versa. Consequently, the number of configurations of an  $N \times N$  point-to-point switch is equal to the number of permutations of  $\{1, \dots, N\}$ , that is, to  $N!$ . Proposition 1 implies that the number of crosspoints of such a switch is at least  $\log_2(N!) \approx N \log_2 N$ , using Stirling's approximation:  $N! \approx (N/e)^N$ , where  $e = 2.718$ . It is therefore not possible to build an  $N \times N$  point-to-point switch with fewer than  $N \log_2 N$  crosspoints. We will use this

lower bound to evaluate some designs and also to explain the degree of blocking in switches with fewer than this number of crosspoints.

The second example is the  $N \times N$  multipoint switch (see bottom of Figure 12.1). This switch can connect any input link to any set of output links. The only restriction on the configurations is that different input links must be connected to different output links. (This kind of switch has multicast capability, in which an input link is connected to several output links. Multicast is useful for videoconferencing and for video distribution.) To calculate the number of switch configurations, observe that each output link can choose arbitrarily the input link to which it is connected. Thus, this switch has  $N^N$  possible configurations. Accordingly, the switch must have at least  $N \log_2 N$  crosspoints. (The  $N \times N$  crossbar is a multipoint switch, too. But it has  $N^2$  crosspoints, much larger than the lower bound of  $N \log_2 N$ .) In summary, an  $N \times N$  point-to-point or multipoint switch requires at least  $N \log_2 N$  crosspoints.

We conclude with a consideration of the other complexity measures using the model of a switch as an arrangement of crosspoints. Suppose the switch has  $m$  crosspoints and suppose it can achieve  $n$  different configurations. Whenever a particular configuration is desired, the switch controller must set the state of each crosspoint (open or closed). The time needed for this is the setup time of the switch. Whenever an existing input-output connection is terminated or a new connection is initiated, the switch configuration is changed. Clearly, satisfactory switch operation requires that the “holding time” of a configuration, the time for which the configuration is unchanged, must be much larger than the setup time. Thus, a switch with a setup time of several milliseconds can support telephone connections that last several seconds or minutes. But it would not be suitable as a packet switch with packet durations of a few microseconds.

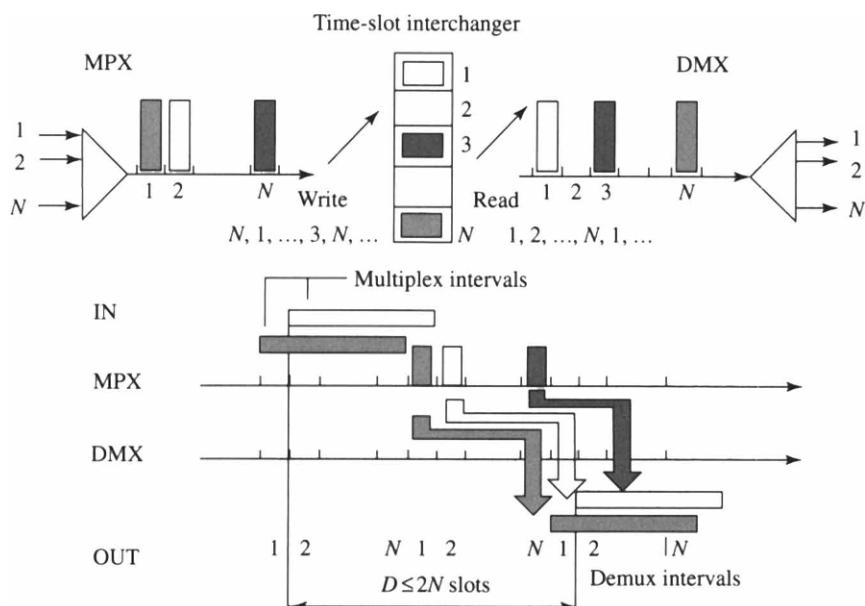
A switch with  $N$  input lines each of which has a capacity of  $b$  bps is said to have a throughput of  $N \times b$  bps. Observe that the speed of the bit stream inside the crossbar of Figure 12.1 is the same as that of the input line. Thus the internal bit rate of this switch is also  $b$  and does not increase with the number of input lines  $N$ . This is in contrast with the time-division switch studied in section 12.2, in which the internal speed equals the throughput. One may be able to reduce the internal speed by a serial-to-parallel conversion. For instance, by first converting the input stream into 16-bit words and arranging the crossbar as 16 bit-level crossbars in parallel, the speed inside the switch is reduced to  $b/16$  bps. The bit stream in each output line is reconstructed by the reverse parallel-to-serial conversion. This introduces more complexity in the switch controller, and the serial-to-parallel conversion introduces some delay. But the lower switch speed may significantly reduce the switch cost.

## 12.2 TIME- AND SPACE-DIVISION SWITCHING

In this section we study the two important principles of time-division and space-division switching.

The telephone network uses time-division switches. Figure 12.2 illustrates the operations of such a switch. The top part of the figure shows  $N$  input signals that arrive on  $N$  different links. These signals are periodic bit streams that must go out on different output links. In the figure it is assumed that the signal arriving on link 1 must go out on link  $N$ , that arriving on link 2 must go out on link 1, ..., and the signal arriving on link  $N$  must go out on link 3.

The time-division switch has three parts: a time-division multiplexer (MPX), a time-slot interchanger, and a time-division demultiplexer (DMX). The MPX first multiplexes the  $N$  incoming signals. That is, it divides time into



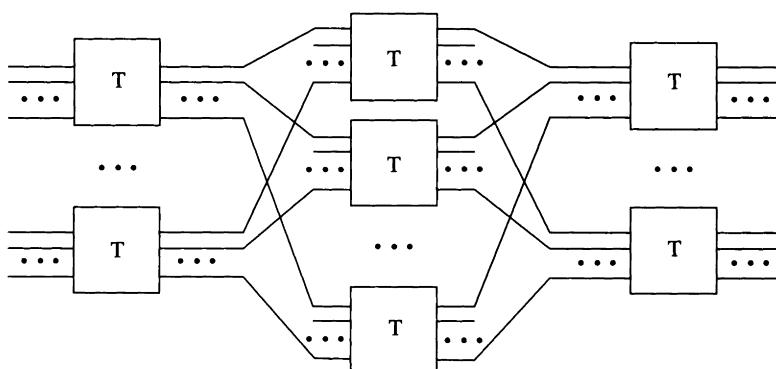
**FIGURE**

A time-division switch consists of a time-division multiplexer (MPX), a time-slot interchanger, and a time-division demultiplexer (DMX).

slots and allocates them to the  $N$  incoming signals in a round-robin manner. The multiplexed signals arrive at the slot interchanger, which writes the successive slots into  $N$  distinct buffers in the order  $N, 1, \dots, 3, N, 1, \dots, 3, N, 1, \dots$ . The order is determined by the switch configuration. The output line of the time-slot interchanger then reads the  $N$  buffers in the order  $1, 2, \dots, N, 1, 2, \dots, N, \dots$ . The output of the slot interchanger is then demultiplexed into  $N$  different signals that are sent to the output links. As is seen in the figure, this switch implements the desired connections between input and output links.

The bottom part of the figure shows the delays of the signals in a time-division switch. The timing diagram traces the evolution of signals through the three stages of the switch. From the diagram we can see that the delay from the time a signal arrives on an input link until it is placed on an output link is at most the duration of two frames of the time-division multiplexer. (A frame is a sequence of  $N$  slots.) Observe that the bit rate of the time-slot interchanger is the same as the throughput of the switch, that is,  $N \times b$  bps, where  $b$  is the bit rate of each link. Therefore the throughput of a time-division switch is limited by the maximum speed of the electronic components.

To achieve greater throughput than possible by time-division switching alone, one can combine it with space division. While a time-division switch separates signals in time, a space-division switch separates the signals in space. The simplest space-division switch is the crossbar of Figure 12.1. We may combine time-division and space-division switches as in Figure 12.3. The boxes labeled T are time-division switches. The internal speed in this combination



12.3

FIGURE

Space-division and time-division switching can be combined to produce large switches.

is the same as that in the time-division switches and does not grow with the number of input ports. Such a combination also enables the switch designer to build a large switch from small modules, as we study next.

### **12.3 MODULAR SWITCH DESIGNS**

In this section we explain how to build large switches from small modules in such a way that the switch has specified connectivity properties. We first examine one class of modular designs, called *Clos networks*, that will form the basic structure of large modular switches.

A Clos network is a collection of switching nodes arranged in a network as in Figure 12.4. The network is composed of three stages of switches. The first two stages are fully connected, that is, each node in the first stage is connected to every node in the second stage. The second and third stages are similarly fully connected. (There is no direct connection between nodes in the first and third stages.) The input (leftmost stage) switches all have the same number of input lines. The output (rightmost stage) switches all have the same number of output lines. Thus, a Clos network is fully specified by five integers:  $(IN, N_1, N_2, N_3, OUT)$ . The network in the figure is Clos  $(3, 3, 5, 4, 2)$ .

We study the nonblocking properties of switches. A switch is said to be *nonblocking* if all the one-to-one connections are compatible. That is, the switch can have all the point-to-point configurations between its inputs and its outputs. A nonblocking switch can be either *strictly nonblocking* (SNB) or *rearrangeably nonblocking* (RNB). The switch is SNB if any new connection from a free input link to a free output link can always be made without having to modify ongoing connections. Otherwise, the switch is RNB.

There is a remarkably simple characterization of SNB and RNB Clos networks, which we state as a theorem.

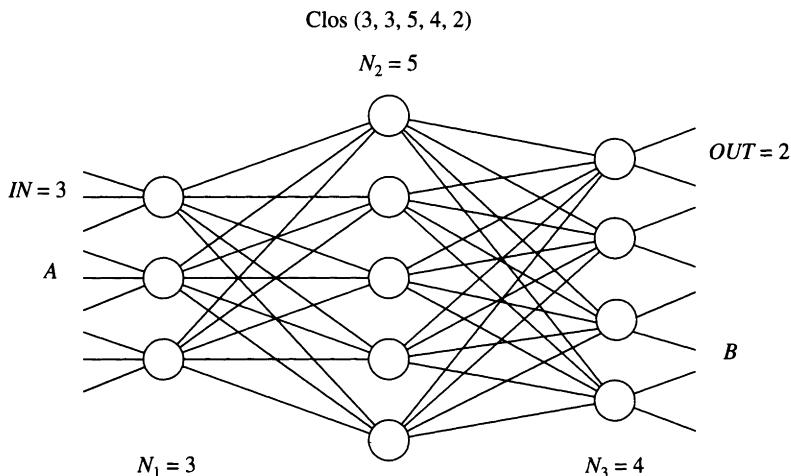
**Theorem 12.3.1** A Clos network built from SNB modules is itself SNB if

$$N_2 \geq IN + OUT - 1. \quad (12.1)$$

That condition is necessary if  $N_1 \geq OUT$  and  $N_3 \geq IN$ .

The form of this result is not surprising: the switch is SNB if it has enough paths to connect input switches to output switches, that is, if  $N_2$  is large enough.

The proof of Theorem 12.3.1 is as follows. (See Figure 12.4.) First we suppose that inequality (12.1) is satisfied, and we show that the Clos network

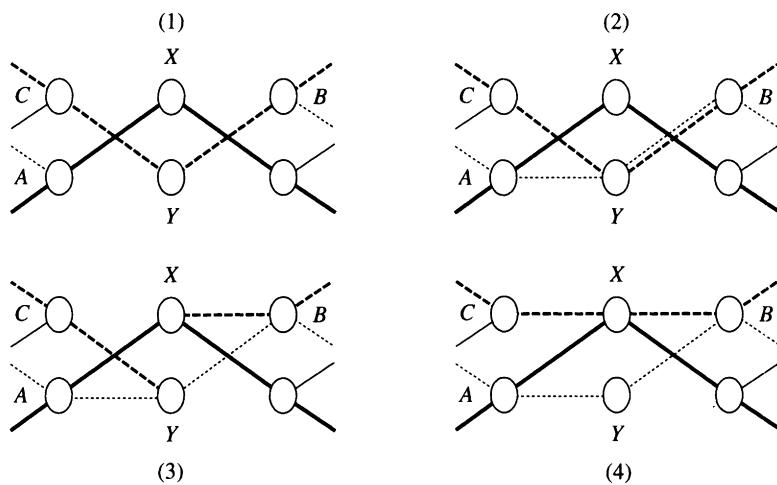
**12.4**

A Clos network is fully specified by  $(IN, N_1, N_2, N_3, OUT)$ .

**FIGURE**

is SNB. To show that, assume that a new connection must be set up from input node  $A$  to output node  $B$ . We must prove that this new connection can be made without having to rearrange existing connections. When a new connection from  $A$  to  $B$  is requested, there can be at most  $IN - 1$  input links busy at switch  $A$  and at most  $OUT - 1$  output links busy at switch  $B$  (not including the new connection). Thus,  $A$  is connected to at most  $IN - 1$  middle switches and  $B$  to at most  $OUT - 1$  middle switches, for a total of at most  $IN + OUT - 2$  switches. Since there are  $N_2 \geq IN + OUT - 1$  middle switches, there must be one middle switch that is connected neither to  $A$  nor to  $B$ . Therefore, the new connection can be made by connecting that middle switch to  $A$  and to  $B$ .

To prove the converse, assume that the switch is SNB. We will show that inequality (12.1) must be satisfied. Consider the worst-case situation when switch  $A$  is connected to  $IN - 1$  output switches excluding  $B$  and when  $B$  is connected to  $OUT - 1$  input switches excluding  $A$ . In that situation,  $A$  is connected to  $IN - 1$  middle switches and  $B$  is connected to  $OUT - 1$  other middle switches. The switches  $A$  and  $B$  are connected to different middle switches since  $A$  and  $B$  are not connected together. Thus,  $IN + OUT - 2$  middle switches are connected either to  $A$  or to  $B$ . Assume that a new connection must then be set up between the free input link of  $A$  and the free output link of  $B$ . This new connection is possible only if there is a middle switch that is not yet connected either to  $A$  or to  $B$ , which requires that  $N_2 \geq IN + OUT - 1$ , as had to be shown.



**FIGURE**  
12.5

A new connection from  $A$  to  $B$  can be established by rerouting the connection through  $Y$ .

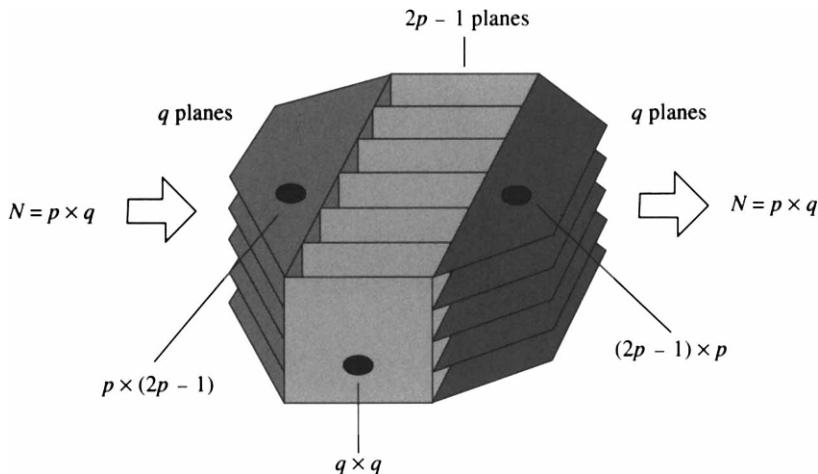
We now obtain the condition for a Clos network to be RNB (rearrangeably nonblocking).

**Theorem 12.3.2** A Clos network built from RNB modules is itself RNB if and only if

$$N_2 \geq \max\{IN, OUT\}. \quad (12.2)$$

Instead of providing a formal proof of this theorem, we illustrate, using Figure 12.5, the rearrangements that enable the switch to set up a new connection from  $A$  to  $B$ . When the connection is requested, in the top-left panel of the figure, there is one existing connection from  $A$  and one existing connection from  $C$  to  $B$ . There is no middle switch that is not already connected either to  $A$  or to  $B$ . Starting with  $A$ , we note a new connection could be made to some middle switch,  $Y$  (since  $N_2 \geq IN$ ), as in the top-right panel. The existing connection between  $C$  and  $B$ , which goes through  $Y$ , is moved to go through  $X$ . The bottom-right panel shows the configuration after the new connection is set up.

We can use Theorem 12.3.1 to construct an  $N \times N$  strictly nonblocking switch (SNTB) with  $N = p \times q$ . The construction is shown in Figure 12.6. The input consists of  $q$  parallel planes. Each of these planes is a  $p \times (2p - 1)$  SNTB input switch. The input planes are attached to  $2p - 1$  parallel  $q \times q$  SNTB



12.6

Recursive construction of a strictly nonblocking switch.

FIGURE

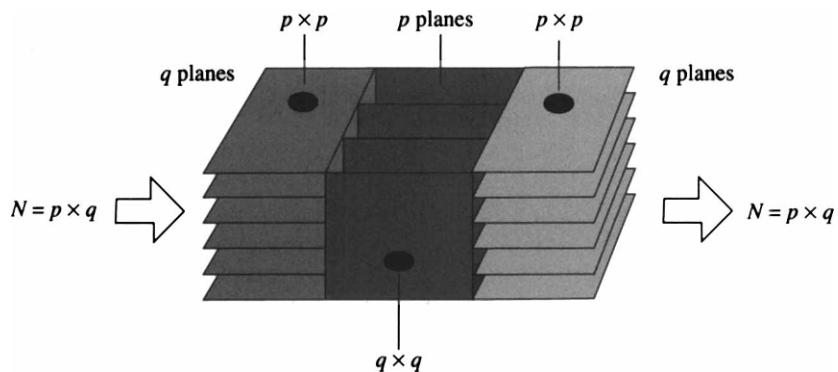
intermediate switches. These  $q \times q$  planes are in turn attached to  $q$  parallel  $(2p - 1) \times p$  SNB output switches.

We can view each input plane as being a first-stage switch in a Clos network, each middle plane as a middle-stage switch, and each output plane as an output switch. Indeed, the first two stages are fully connected and so are the last two stages, so that the configuration is a Clos network. Thus, this modular switch is a  $(p, q, 2p - 1, q, p)$  Clos network, in the notation introduced earlier and in Figure 12.4, and by Theorem 12.3.1 this switch is strictly nonblocking.

To appreciate the advantage of this modular construction, let us assume that each of the switching modules is implemented as a crossbar. In that case, each input and output plane has  $p \times (2p - 1)$  crosspoints and each middle plane has  $q \times q$  crosspoints. Consequently, the complete  $N \times N$  switch has  $2p(2p - 1)q + q^2(2p - 1)$  crosspoints. But if the  $N \times N$  switch had been implemented as a crossbar, it would have  $N^2 = p^2q^2$  crosspoints. As a numerical illustration, when  $p = q = 100$ , the modular design has only 4% of the crosspoints of a crossbar.

Figure 12.7 shows another modular switch design except that it is RNB instead of SNB. This design is based on Theorem 12.3.2.

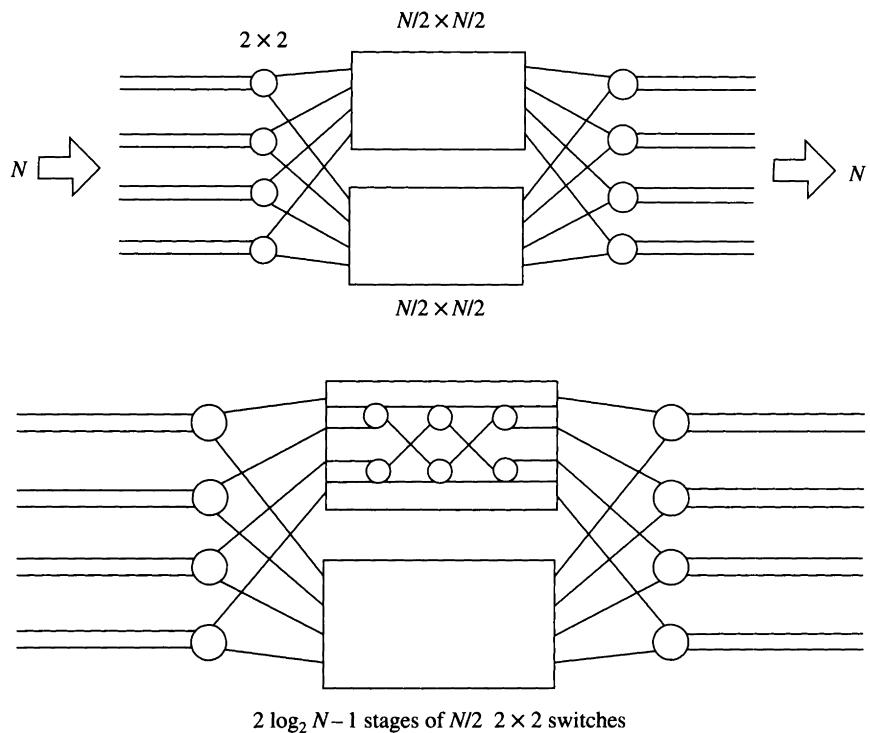
Using Figure 12.8 we explain the construction of a Benes network. This is an RNB switch built in a specific recursive way. The switch is  $N \times N$ , where  $N$  is a power of 2. The top of the figure shows the first step of the construction, a  $(2, N/2, 2, N/2, 2)$  Clos network. According to Theorem 12.3.2, the switch is RNB when its two modules are RNB. The next step is explained in the bottom of



12.7

FIGURE

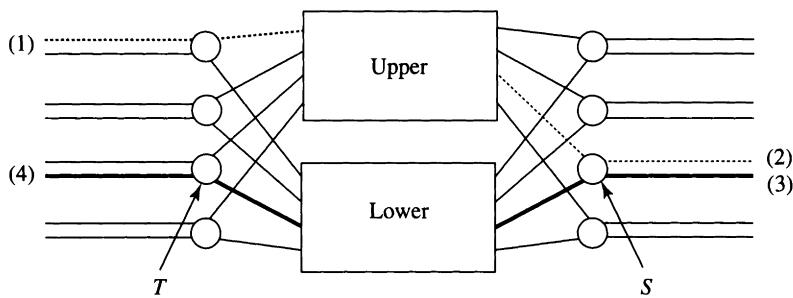
Recursive construction of a rearrangeably nonblocking switch.



12.8

FIGURE

Recursive construction of a Benes switch.



12.9

Routing in a Benes switch can be done recursively.

FIGURE

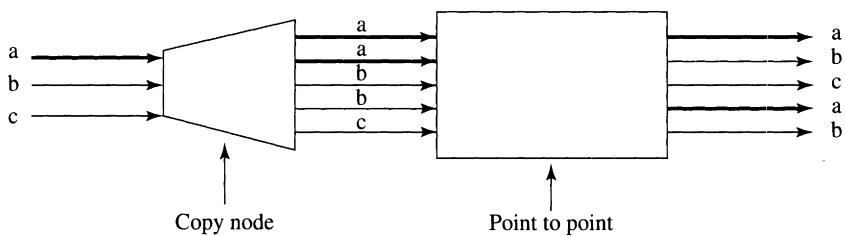
the figure: each of the two  $(N/2) \times (N/2)$  modules of the Benes switch is again decomposed as in the first step. Continuing in this way yields a switch with  $2 \log_2 N - 1$  stages each of  $N/2 2 \times 2$  switches such that the total number of crosspoints is approximately  $4N \log_2 N$ . So the complexity of the Benes network is four times the minimal complexity (see Proposition 12.1.1, above). Thus, the complexity of the Benes network has the optimal order.

Because the Benes switch is not SNB, the request for a new input-output connection may require rerouting existing connections. Routing in a Benes network can be performed with an algorithm that can be explained using Figure 12.9. Assume that initially no connections are made and let  $C$  be the desired set of input-output connections.

- ◆ Step 1. Pick an input-output pair in  $C$  not already selected. (Terminate if none exists.)
- ◆ Step 2. Connect the input (1) to the output (2) of  $S$  (say) via the upper block  $U$ .
- ◆ Step 3. If the other output (3) of  $S$  desires connection to the input (4) of  $T$  (say), connect via the lower block  $L$ . Then return to Step 1.

Apply the algorithm recursively to  $U$  and  $L$ . This algorithm shows that the simplicity of the modular construction of a Benes network is matched by a simple routing algorithm.

So far we have discussed the modular construction of point-to-point switches. A similar construction can be carried out for multipoint switches. As Figure 12.10 shows, one implementation of a multipoint switch is a copy node followed by a point-to-point switch. The advantage of such an implementation is that modular designs exist for copy nodes.



**12.10**  
**FIGURE**

A multipoint switch can be built by a copy node followed by a point-to-point switch.

## 12.4

## PACKET SWITCHING

The configuration of a circuit switch must be changed every time a new connection or call is to be established or an existing connection is to be torn down. A packet switch must make a forwarding decision every time a new packet arrives. Since the duration of a circuit-switched phone call lasts several minutes whereas a new packet may arrive every few  $\mu\text{s}$ , the frequency of changes in the switch is orders of magnitude greater in the case of packet (and virtual circuit) switching than in the case of circuit switching. This difference has a major impact on switch design and performance. We first compare the functions that the two types of switches must perform.

In circuit switching, during the connection setup phase, the network assigns to the admitted call a route through the network and an idle circuit in each link along the route. Usually, a link's capacity is divided into many circuits, by time-division multiplexing. But let us suppose, for simplicity, that each link carries a single circuit. The route assigned to a call then requires each switch along the route to connect a particular idle incoming link  $i$  and a particular idle outgoing link  $j$ . Data arriving on link  $i$  must be transferred to link  $j$  for the entire duration of the connection. The switch controller must therefore maintain lists of idle incoming and outgoing links and the current configuration (i.e., a list of currently connected input-output link pairs). These lists serve two purposes: internally, to configure the switch and, externally, to notify the network's call-admission and call-routing procedures about the availability of idle links. These lists and the switch configurations will change whenever a new connection is set up or an existing one is torn down. The interval between changes is the call holding time, which is on the order of minutes or seconds. Note also one important additional feature of circuit switching. Once a connection between links  $i$  and  $j$  has been established, data coming in

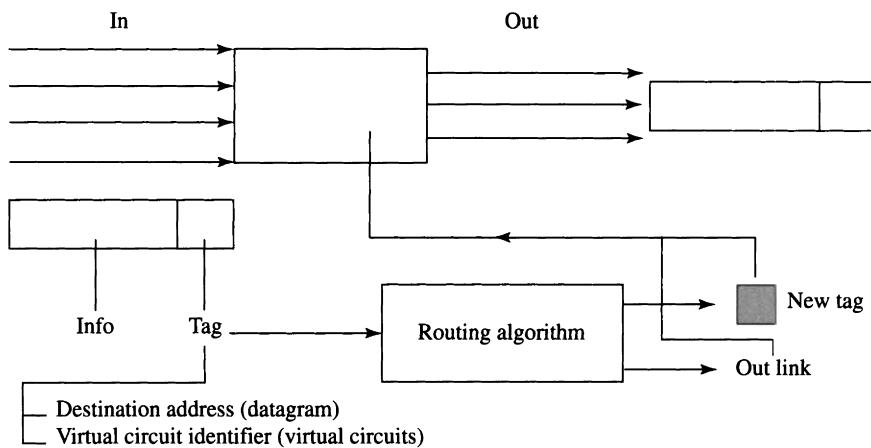
over link  $i$  is transferred out with negligible delay to link  $j$ , so no buffering is needed inside the switch and a small amount of buffering is needed at the input ports.

Packet switching differs from circuit switching in several ways. First, two packets arriving consecutively on the same incoming link may be destined for different outgoing links. In datagram networks, this can happen because successive packets are routed independently even if they have the same destination; in virtual circuits this can happen because two consecutive packets belonging to different virtual circuits and arriving on the same incoming link may leave the switch on different outgoing links. The forwarding decision in a packet switch thus changes every time a packet arrives, so the time duration between decisions can be as short as a fraction of  $\mu\text{s}$  or  $\text{ms}$ . (For example, a 53-byte ATM cell takes less than  $0.7 \mu\text{s}$  to transmit at 622 Mbps.) Second, the controller must examine each incoming packet to obtain either the destination address identifier (in the case of datagrams) or the virtual circuit identifier (in the case of ATM cells). Third, the controller must determine the outgoing link(s) and possibly a new header for that packet. Fourth, an incoming packet may have to be replicated and sent to multiple output ports. This possibility arises in multicast applications where a number of users get copies of packets that a source sends.

Finally, unlike in circuit switching, during a short time interval it may happen that more packets are destined over the same outgoing link than can be transmitted given that link's capacity. In that event, the switch will have to buffer some packets. (A packet may be buffered also if there is contention within the switch.) This buffering function is absent in circuit switching.

Figure 12.11 illustrates the key ideas in packet-switching schemes. As the incoming packet is being read into a buffer, the forwarding algorithm of the controller processes the destination address or virtual circuit number or routing tag (all called *tag* in the figure). The algorithm calculates the outgoing link and, possibly, a new tag for the outgoing packet.

Consider a virtual circuit-switched network. Then the routing algorithm maintains a table with entries  $(VCI_{in}, Port_{in}, VCI_{out}, Port_{out})$ .  $VCI_{in}$  is an entry in the incoming packet tag that is replaced by  $VCI_{out}$  in the outgoing packet tag.  $VCI_{out}$  may be different from  $VCI_{in}$ , for reasons to be explained shortly. When a new virtual circuit is being set up, the network call-admission procedure selects a route through the network, and a new entry is created in every table along the path. When the virtual circuit is torn down, the switches along the route are informed, and the corresponding entry is deleted. (The procedure for selecting a route may be similar to that used in telephone networks, or it may use a shortest-path algorithm. See section 6.2.3.)

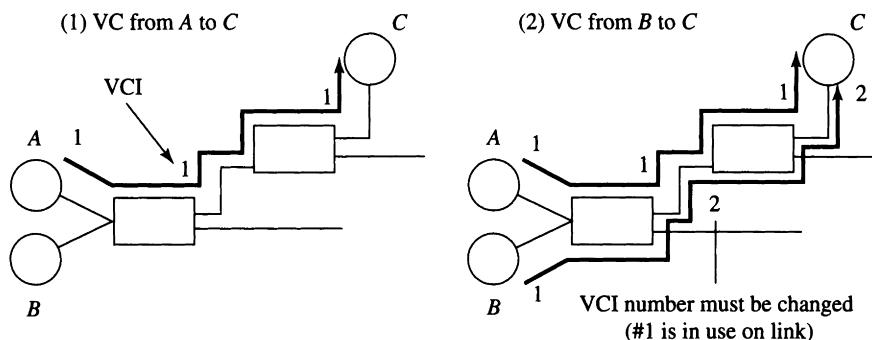


**12.11**  
**FIGURE**

The tag of an incoming packet is decoded to determine the output link and the new tag.

In case of a datagram network, the tag contains a destination address identifier instead of a VCI, and the routing algorithm maintains a table with entries (destination address,  $Port_{out}$ ). This table may be static or dynamic. In the dynamic case, the entry corresponds to the current shortest path from the switch to the destination. As the traffic pattern changes, so does the queuing delay in each link. As a result the shortest path from source to destination changes. The shortest path is periodically reestimated by the router by a separate algorithm. (A shortest-path algorithm is described in section 4.3.)

Having selected a route, Figure 12.12 illustrates how a VCI is assigned to the new connection. A new VCI is assigned by the source. A source cannot simultaneously use the same VCI for two different connections originating there. But different sources may use the same VCI. When two different sources assign the same VCI and the two virtual circuits go through the same switch, there is a potential conflict. To prevent this conflict, when the second virtual circuit is being set up, the first switch in common to the two paths will assign a new outgoing VCI to the second virtual circuit. This can be implemented by transmitting a special packet along the route of the new virtual circuit being set up. Thus, for example, in the figure, VCI #1 has already been assigned to the connection between A and C, when source B initiates a new virtual circuit with the same VCI. When the special connection setup packet arrives at the switch common to both routes, the conflict is noted, and the switch assigns the new (unused) VCI #2 to the new virtual circuit.



12.12

The VCI is unique within each link. In case of conflict, a new VCI is assigned.

FIGURE

In virtual circuit switching, each packet carries a VCI, whereas in datagram switching, each packet carries a destination address. The destination address must be unique throughout the network, whereas a VCI must be unique only within each link. Therefore packets in virtual circuit networks will have a shorter header than packets in datagram networks. However, there is an extra overhead in setting up and tearing down a virtual circuit connection that is absent in datagrams.

**Searching Routing Tables** Searching a routing table to locate the forwarding decision that corresponds to a given packet or cell may be a bottleneck on the throughput of a switch. We examine a few strategies that switches use to perform such a search quickly.

To compare the search strategies, consider a routing table that specifies the forwarding decisions for the routing tags of incoming packets. Assume that the routing tags have  $N$  bits. For instance, in IP routing, the routing tag is the IP address of the destination, so that  $N = 32$ . In an ATM network, the routing tag is generally the VPI, so that  $N = 16$  (the routing could also be based on the VCI). In a VLAN, the forwarding decision is based on the LAN ID part of the VLAN tag,  $N = 16$ . In MPLS, the proposed routing tag has  $N = 20$  (for the circuit identifier part of the tag) but only the active connections have an entry in the look-up table in contrast with IP routing tables that have an entry for the IP domains. Assume also that the forwarding decision is specified by  $M$  bits. For instance, in IP-routing, the  $M$  bits specify the port number and the IP address of the next router.

The problem is then to locate an  $N$ -bit entry in a table. Four strategies are used: direct addressing, associative memory, tree search, and hashing. We explain these strategies next.

When using direct addressing, the routing table has  $2^N$  entries of  $M$  bits: one entry for every possible routing tag. The routing table is very easy to search but is very large. For instance, when  $N = 32$ , the table has about  $4 \times 10^9$  entries of at least 4 bytes each. The table requires a memory of 16 Gbytes.

An associative memory, or contents-addressable memory (CAM), is designed so that it returns the address of the location that contains a particular  $N$ -bit word. For instance, the memory may contain  $2^K$  entries of  $N$  bits. When a packet arrives, the controller consults the CAM to find the entry that corresponds to the routing tag of the packet. This entry is designated by  $K$  bits, which can then be used to locate the  $M$  bits that specify the routing decision. CAM memories are expensive and are only available in relatively small sizes.

A tree search strategy may be designed as follows. We group the  $N$  bits  $n$  by  $n$ , where  $N = n \times m$ . Let  $b_1, b_2, \dots, b_m$  be the  $m$  words of  $n$  bits that make up the  $N$ -bit address. To each of the  $2^n$  possible values of  $b_1$  we associate either a table or a null pointer. Each of these tables is a list of  $2^n$  pointers to other tables or null pointers that correspond to the possible values of  $b_2$ , and so on. We traverse these tables until we reach either a null pointer or an entry that corresponds to the  $N$ -bit word. If no pointer is the null pointer, then this search corresponds to a balanced tree whose nodes have  $2^n$  children. Of course, a full tree has  $2^N$  terminal leaves. To see how the tables can be filled up, imagine that we present  $N$ -bit routing tags together with the corresponding forwarding decision to the algorithm. For every new first word  $b_1$ , the algorithm prepares a pointer to a free memory location. The same procedure continues for the other words. The entry of the last table is filled up with the routing decision.

Hashing can be used effectively as follows to implement a fast search through a large table. One chooses a function  $g : \{0, 1\}^N \rightarrow \{0, 1\}^K$ . We use a memory with  $2^K$  entries of  $N + M$  bits. Let  $x(1), x(2), \dots$  be the entries that we must add to the table. We enter  $x(1)$  at the address pointed to by the  $K$ -bit word  $g(x(1))$ , and we add the  $M$ -bit forwarding decision to that address. We continue in this way until we have to enter an entry  $x(i)$  such that the  $K$ -bit word  $g(x(i))$  points to an address that is already used. In that case, we look for the first address after  $g(x(i))$  that is not used yet. We continue in this way until we have entered all the entries. To locate an entry  $x$  in the table, we look at the address  $g(x)$ . If the routing tag at that address is  $x$ , then we are done. If not, we look at the subsequent entries until we find  $x$ . Assuming that the different values  $g(x(1)), g(x(2)), \dots$  are uniformly distributed in  $\{0, 1\}^K$ , we can estimate the number of entries we need to search.

We now introduce four packet switch (PS) designs: the distributed buffer, the shared buffer, the output buffer, and the input buffer. The designs differ in the way a packet is routed through the switch and in the way it is buffered when there is contention. Accordingly, their performance, measured in throughput, delay, or buffer requirements, varies. The distributed buffer design is the only one of the four that is modular in construction. Therefore, it can be scaled to accommodate a large number of ports. The other three designs cannot be scaled without increasing the speed inside the switch; they are more suitable for local area switches.

## 12.5

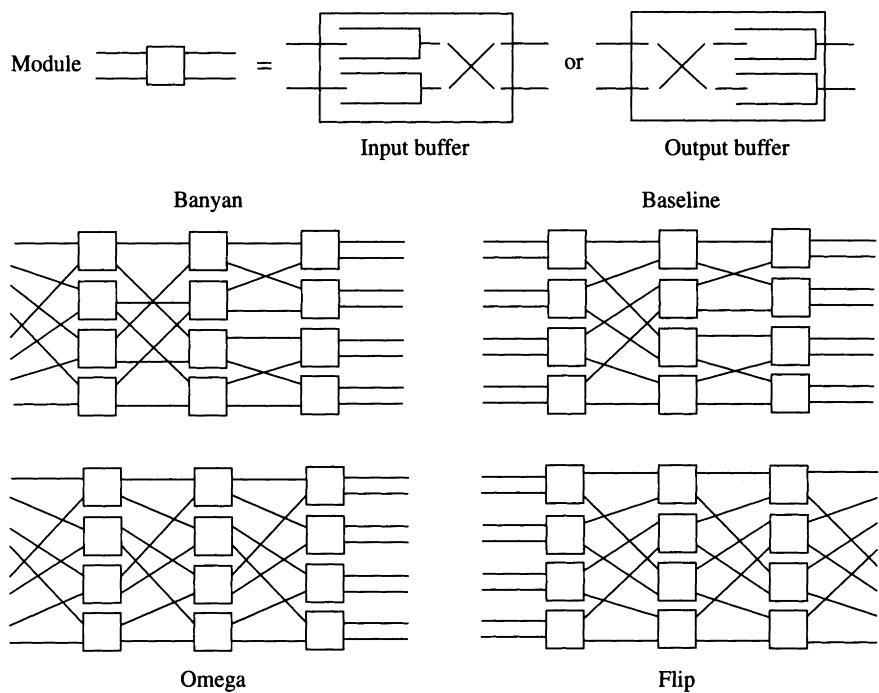
## DISTRIBUTED BUFFER

The distributed buffer switch (DBS) designs have a modular construction like, say, the Benes network. However, there is an important difference. The Benes network realizes the maximum number of possible switch configurations for a given number of crosspoints, but reconfiguring the switch is complicated. The DBS designs, on the other hand, can realize far fewer configurations for the same number of crosspoints, but routing a packet through the switch—so-called tag-based or self-routing—is especially simple. The Benes network is optimized for circuit switching (where reconfiguration is relatively infrequent), the DBS designs are optimized for packet switching.

Figure 12.13 illustrates four “delta” networks. These are arrangements of  $2 \times 2$  crosspoint modules. If there are  $N = 2^n$  input and output ports, there are  $N/2$  rows and  $n = \log_2 N$  stages, for a total of  $N/2 \log_2 N$  modules, hence  $2N \log_2 N$  crosspoints. (Compare these arrangements with the Benes switch with twice as many stages.) Each module contains its own buffer—hence the name *distributed buffer*. (The module may have an input or output buffer, as shown, or it may have no buffer at all.) We will first discuss how all these arrangements permit rapid self-routing of a packet or cell (we use both names interchangeably) from any input port to any designated output port. The different arrangements vary in the way the route through the switch from input port to output port is selected. We will then discuss buffering.

A cell arrives at a switch input port. (See Figure 12.14.) The cell consists of two parts: data and tag. The tag contains the destination address (if the cell is a datagram) or the VCI (if the cell is part of a virtual circuit connection). This tag is read by the switch controller, which determines the output port to which the cell must be routed.

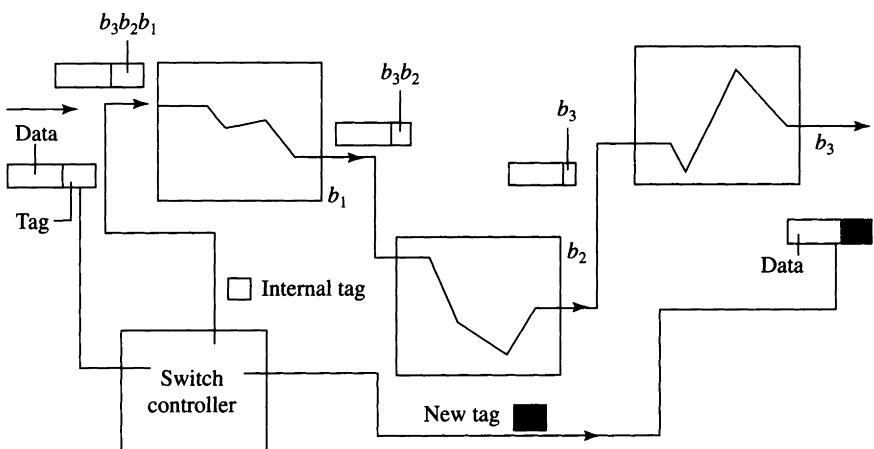
Having determined the output port, the switch controller replaces the arriving tag by an internal tag (or prepends the internal tag) that designates



12.13

**FIGURE**

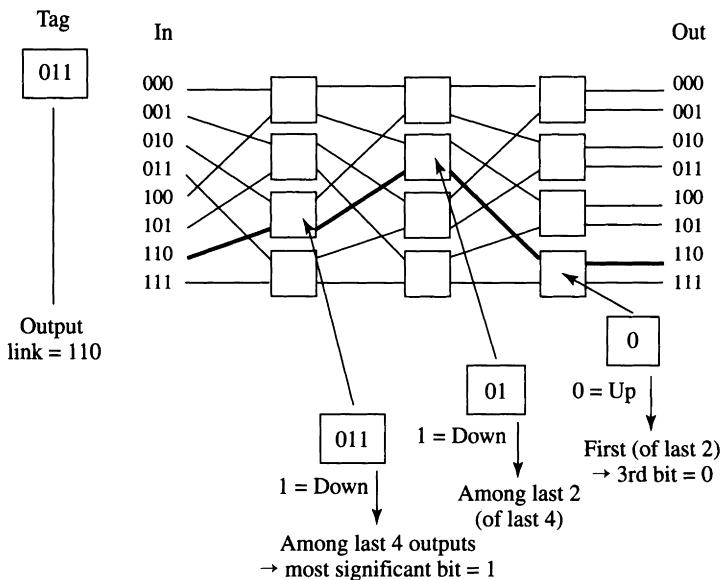
Four variants of distributed buffer switches. Each  $2 \times 2$  module contains an input or output buffer.



12.14

**FIGURE**

In self-routing, the tag is decoded bit by bit by each module to determine the route through the distributed buffer switch.



12.15

Self-routing in an omega network. If the output port is  $ABC$ , the tag is  $CBA$ .

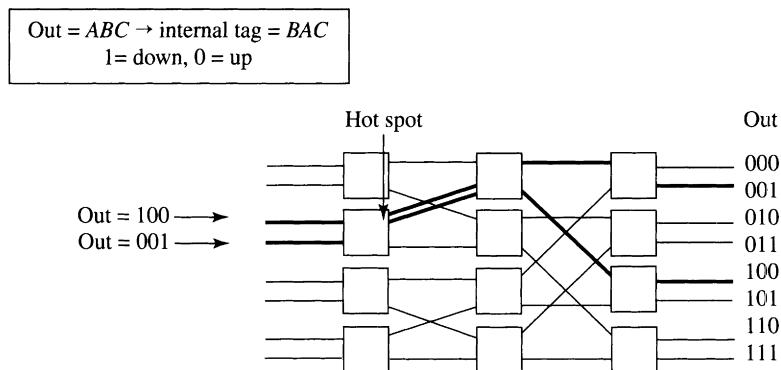
FIGURE

the output port. So this tag is a sequence of  $n = \log_2 N$  bits. These bits are used one at a time by the  $\log_2 N$  stages of  $2 \times 2$  modules to determine whether the cell should go “up” or “down.” In the illustration of the figure,  $n = 3$ , so the tag =  $b_3b_2b_1$ . In the first stage, bit  $b_1$  is examined by the input port and decoded to determine whether the cell goes up or down. Bit  $b_1$  is then stripped. At stage 2, bit  $b_2$  is decoded to determine whether the cell goes up or down, and then stripped. Lastly, in stage 3, bit  $b_3$  is decoded. The cell, stripped of its tag, arrives at the proper output port designated by  $b_3b_2b_1$ .

While the cell is being routed through the switch fabric, the switch controller calculates the new tag from the routing table. It is appended to the outgoing cell.

Figure 12.15 illustrates the omega network, in which the tag is the output port number in reverse order: if the output port is  $ABC$ , then the tag is  $b_3b_2b_1 = CBA$ . For routing, bit 1 is interpreted as down, and bit 0 is interpreted as up. In the bottom of the figure, we explain why this encoding of the output port leads to the correct route. Note that there is a unique route to each output port from each input port; this makes routing simple.

Figure 12.16 illustrates the banyan network. The top of the figure gives the routing. The idea is the same as for the omega network, but the tag-to-route encoding is different: if the output port is  $ABC$ , the tag is  $CAB$ ; again bit 1 is



**12.16**  
**FIGURE**

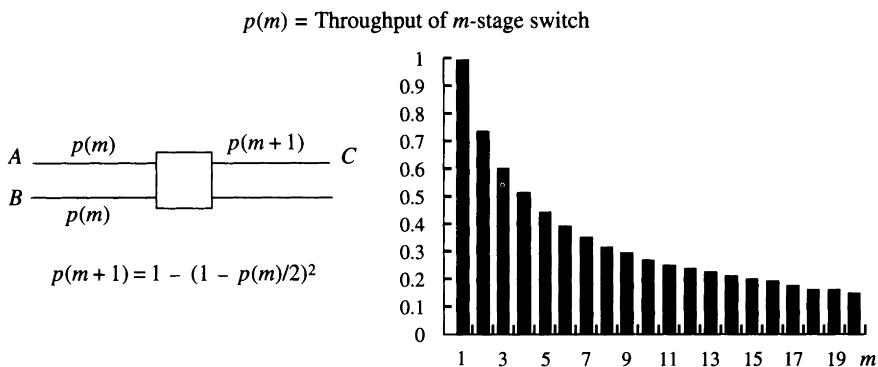
Queuing in a banyan network: two packets destined for different ports may contend for transfer along the same internal link causing a queuing delay.

down and 0 is up. The figure shows the routes from the two input ports to two output ports 100 and 001.

Having understood how packets are routed within a switch, we can evaluate the crosspoint complexity of these distributed buffer switches. There are  $N/2 \log_2 N$   $2 \times 2$  modules that each admit two states (crossed or uncrossed). Hence the switch has  $N^{N/2}$  states. Each state corresponds to a particular configuration. Hence these arrangements can realize only about  $(N!)^{1/2}$  permutations, out of a total of  $N!$  permutations. (By comparison, a Benes network with twice as many crosspoints can realize all permutations.) Thus, distributed buffer switches are not nonblocking, that is, even when the cells present at all input ports are going to different output ports (a permutation), these cells may be unable to travel simultaneously. This happens whenever they share the same path at some intermediate module, and when there is contention, one or more cells will have to be buffered. (We see in Figure 12.16 that cells destined for output ports 100 and 001 contend for the same intermediate module and so one of these cells must be buffered.) Thus, rapid self-routing is achieved at the cost of blocking of packets at intermediate stages inside the switch with a resulting need to buffer. Intermediate modules where queues build up are called *hot spots*.

### 12.5.1 Impact of Hot Spots

We will calculate the reduction in throughput of a DBS due to hot spots beginning with the simplest case where the modules have no buffers. When two cells are routed through the same link, only one of them can get through and



12.17

FIGURE

If there is no buffering, the throughput declines rapidly with the number of stages.

the other cell is dropped. Suppose cells arrive at each time slot, one per switch input port. The destination of each cell is random, uniformly distributed over the output ports, and different cells are independent.

Let  $p(m)$  be the probability that a cell is forwarded over a link of the DBS after going through  $m$  stages, in one slot. So  $p(0) = 1$ , since an external cell arrives at each input link in each time slot. Consider a link in a module after  $m + 1$  stages, and one slot. Suppose it is link  $C$  in Figure 12.17. By definition, the probability that a cell is not forwarded over link  $C$  is  $1 - p(m + 1)$ . This event occurs if and only if no cell is routed to  $C$  from either link  $A$  or  $B$ . The probability that a cell is routed from link  $A$  to  $C$  is  $0.5p(m)$ , so the probability that a cell is not routed from  $A$  to  $C$  is  $[1 - 0.5p(m)]$ . This is also the probability that a cell is not routed from  $B$  to  $C$ . Since these two events are independent, the probability that no cell is routed to  $C$  from  $A$  or  $B$  is  $[1 - 0.5p(m)]^2$ , and so  $1 - p(m + 1) = [1 - 0.5p(m)]^2$ . This gives the recursion

$$p(m + 1) = 1 - [1 - 0.5p(m)]^2, \quad p(0) = 1,$$

which yields  $p(0) = 1$ ,  $p(1) = 0.74$ ,  $p(2) = 0.62$ ,  $p(3) = 0.52$ ,  $p(4) = 0.45$ , and so on. The throughput through a link after  $m$  stages is  $p(m)$ . (The remainder,  $1 - p(m)$  cells, are lost due to conflict.) Figure 12.17 shows that the throughput decreases rapidly as the number of stages increases.

To avoid this decline in throughput, each module must buffer cells when there is contention on its output links. As we indicated in the top of Figure 12.13, the buffer may be at the input or at the output ports of the module. The analysis of the buffer occupancy process is complicated, and we shall make some simplifying assumptions. We first consider the input buffer case.

### 12.5.2 Input Buffers

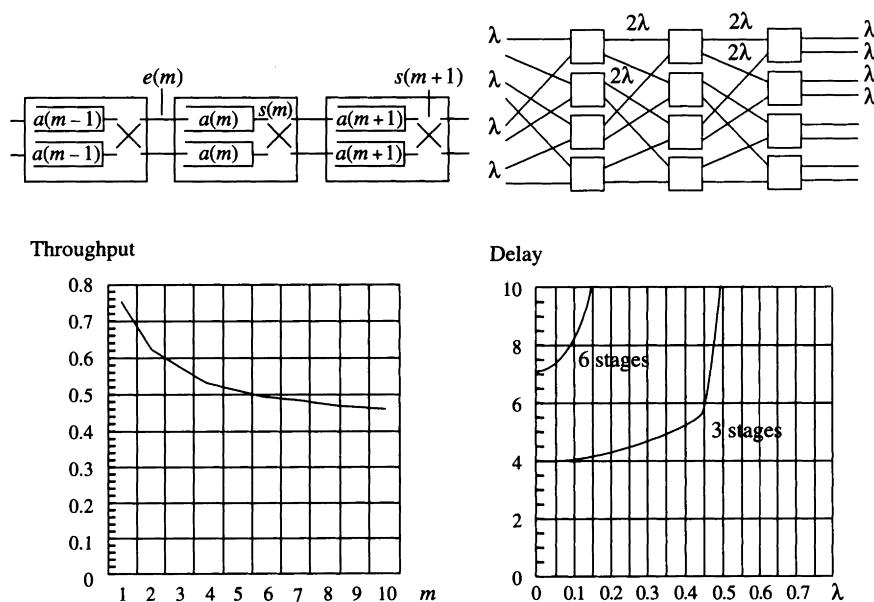
We suppose, as before, that iid (independent, identically distributed) cells arrive at the switch in each time slot and each input port with probability  $\lambda$ , with uniformly distributed destination. Suppose also that each buffer can hold only one cell. Then a buffered cell can move forward over a link if there is no conflict (another cell is not trying to use the same link) and if there is no blocking (the buffer at the end of the link is not full). We now make the simplifying assumption that the buffers are independent and iid per stage. (In fact this is not the case: if a buffer is occupied, it will increase the likelihood that the buffer upstream of it will also be occupied.) Consider the top-left panel in Figure 12.18.

For each buffer in each stage  $m$  and time slot  $t$ , define

$$a_t(m) := P(\text{queue size at end of } t = 0) =: 1 - b_t(m),$$

$$e_t(m) := P(\text{cell ready to enter queue during } t),$$

$$s_t(m) := P(\text{cell can move forward during } t \mid \text{cell is present during } t).$$



12.18

FIGURE

Throughput and delay when there is a single buffer.

Then,

$$\begin{aligned} a_t(m) &= [1 - e_t(m)][a_{t-1}(m) + b_{t-1}(m)s_t(m)], \\ e_t(m) &= 0.75b_{t-1}(m-1)b_{t-1}(m-1) + 2 \times 0.5 \times a_{t-1}(m-1)b_{t-1}(m-1), \\ s_t(m) &= [a_{t-1}(m) + 0.75b_{t-1}(m)][a_{t-1}(m+1) + b_{t-1}(m+1)s_{t-1}(m+1)]. \end{aligned}$$

The first equation says that the buffer is empty at the end of slot  $t$  if there is no arrival during  $t$  and either the queue was empty at the end of slot  $(t-1)$  or there was a cell in the queue and it left during slot  $t$ . The second equation accounts for the likelihood of an arrival depending on whether both buffers attached to its incoming links are occupied, or exactly one of those buffers is occupied. The third equation says that a cell can be forwarded only if there is no conflict with a cell in the second buffer in the same stage  $m$  and if it cannot be blocked by a cell in the buffer in stage  $m+1$ .

To compute the steady-state probabilities, we can drop the index  $t$  and get

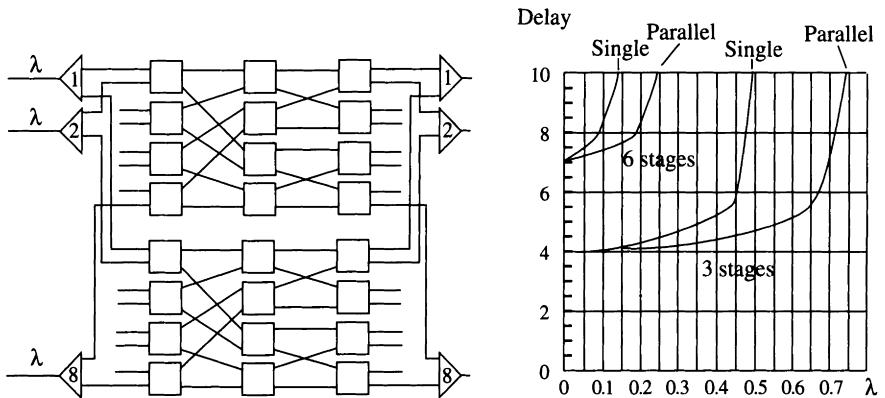
$$\begin{aligned} a(m) &= [1 - e(m)][a(m) + b(m)s(m)], \\ e(m) &= 0.75b(m-1)b(m-1) + 0.5 \times 2 \times a(m-1)b(m-1), \\ s(m) &= [a(m) + 0.75b(m)][a(m+1) + b(m+1)s(m+1)]. \end{aligned}$$

These equations can now be solved numerically, with appropriate boundary conditions:  $e(1) \equiv \lambda$  (a cell arrives with probability  $\lambda$  at stage 1 in each slot) and  $s(n) \equiv 1$  (a cell can leave from each output line). The maximum throughput of the DBS with  $n$  stages is  $b(n)s(n)$  (for  $\lambda = 1$ ), which is plotted in the lower left panel of Figure 12.18. Comparing this figure with Figure 12.17, we can see the improvement in throughput obtained by providing single cell buffers. As expected, the throughput declines with the number of stages,  $n$ , dropping to 40% for  $n = 10$ , or  $2^{10} = 1,000$  input ports.

The delay—the expected number of slots it takes for an incoming cell to reach its output port—can be calculated from the  $s(m)$ . Since  $s(m)$  is the probability that a cell will move forward at stage  $m$  in one time slot, the expected number of slots it will take to move forward is  $1/s(m)$ , and so the delay through an  $n$  stage switch is

$$\sum_{m=1}^n \frac{1}{s(m)}.$$

The probabilities  $s(m)$  (and the other probabilities introduced earlier) depend on the distribution of the destination output port as well as the arrival rate. Figure 12.18 gives a plot of the delay versus arrival rate  $\lambda$  for a worst-case distribution. The top-right panel shows this worst-case distribution for the omega



12.19

## FIGURE

Throughput improves when parallel paths are provided.

network. Notice how traffic arriving on four input ports and destined for four different output ports nevertheless is routed through only two internal links.

The number of conflicts (cells needing to travel over the same internal link) would decrease if cells were provided alternative paths to their destination. One way of doing this is to build parallel networks. Figure 12.19 shows the reduction in delay with two parallel baseline networks. Notice that traffic on the top four incoming ports is routed through four parallel links, two in each network.

So far we have assumed that buffers can hold only one cell. Clearly the performance will improve with larger buffers. Suppose each buffer can hold  $b$  cells. The equations that govern the evolution of the various probabilities can be derived as follows.

For each buffer in each stage  $m$  and slot  $t$ , define

$$a_t(m, i) := P(\text{queue size at end of } t = i),$$

$e_t(m) := P(\text{cell ready to enter queue during } t),$

$s_t(m) := P(\text{cell can move forward during } t \mid \text{cell is present during } t)$ .

Then,

$$a_t(m, i) = a_{t-1}(m, i - 1)e_t(m)(1 - s_t(m)) + a_{t-1}(m, i)[e_t(m)s_t(m)]$$

$$(1 - e_t(m))(1 - s_t(m))] + a_{t-1}(m, i+1)(1 - e_t(m))s_t(m), \quad 0 \leq i \leq b,$$

$$e_t(m) = 1 - \left[ 1 - \frac{1}{2}(1 - a_t(m-1, 0)) \right]^2,$$

$$s_t(m) = \left[ \frac{e_t(m+1)}{1 - a_t(m, 0)} \right] [1 - a_t(m+1, b) + a_t(m+1, b)s_t(m+1)].$$

These equations generalize those for the case where a buffer can hold only one cell. To obtain the first equation, we begin by noting that to have  $i$  cells in the buffer at the end of slot  $t$ , there must be  $i - 1$ ,  $i$ , or  $i + 1$  cells at the end of slot  $t - 1$ . These possibilities account for the three terms in the right-hand side of the first equation. (If  $i = 0$ , the first term is absent, and if  $i = b$ , the third term is absent.) The second equation is based on the observation that there is no arrival into a buffer, which happens with probability  $(1 - e_t(m))$ , if and only if neither of the two buffers attached to its incoming links contains a cell destined for the stage  $m$  buffer being considered, which occurs with probability  $[1 - 1/2(1 - a_t(m - 1, 0))]$ . The third equation says that a cell can be forwarded if there is at least one cell in the buffer ready to enter the queue at the next stage (the probability of this is given by the first term in square brackets) and if it cannot be blocked by a cell in the buffer at stage  $m + 1$  (this has probability given by the second term).

To compute the steady-state probabilities, we can drop the index  $t$  and solve the resulting recursive equations numerically, using appropriate boundary conditions as before. We can then obtain the throughput of a DBS with  $n$  stages as  $s(n)[1 - a(n, 0)]$ . Let  $R(m)$  be the probability that a cell will move forward at stage  $m$  in one slot, so the expected number of slots it takes to move forward is  $1/R(m)$  and the delay or expected number of slots it takes for an incoming cell to reach its output port through an  $n$  stage switch is

$$\sum_{m=1}^n \frac{1}{R(m)}.$$

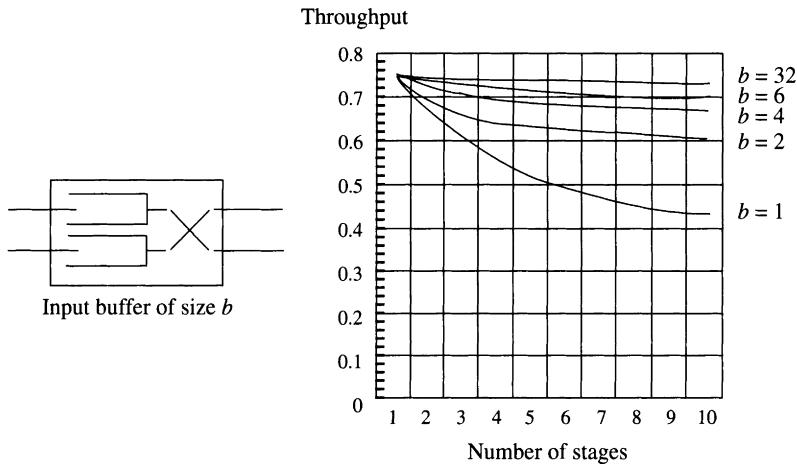
To compute  $R(m)$  we note that if there are  $i$  cells in the buffer, the probability that a particular cell will move forward in one slot is  $s(m)/i$ . Hence,

$$R(n) = s(n) \sum_{i=1}^b \frac{1}{i} \frac{s(n, i)}{1 - a(n, 0)}.$$

Figure 12.20 displays plots of throughput as a function of the number of stages for different values of  $b$ .

### 12.5.3 Combating Hot Spots

We have seen that the performance of a DBS, measured by throughput or delay, is reduced by queuing or hot spots. We consider two techniques to combat hot spots. The techniques are based on insight gained from the following model of how hot spots develop. Consider a DBS with  $N$  input and output ports. Suppose that in each time slot one cell or packet arrives at each input port, destined for



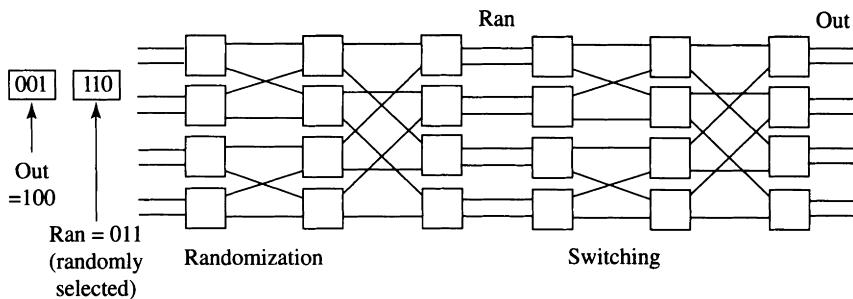
12.20

FIGURE

one of the output ports. Thus the incoming traffic is characterized by a time sequence of “destination”  $N$ -dimensional vectors of the form  $(Port_{out}^1, \dots, Port_{out}^N)$ , where  $Port_{out}^i$  is the output port destination of the cell at the  $i$ th input port. Associated with each destination vector are  $N$  routes through the switch, one from each input port to the corresponding output port, and one packet must be transmitted along each link of each of the  $N$  routes. The total number of packets for any link equals the number of routes through that link and may vary from 0 to  $N$ . If this number is 0 or 1, no packets will be buffered at that link; if the number is 2 or more, packets will have to be buffered. Some sequences of destination vectors will lead to unbalanced traffic (i.e., they cause some links to carry much more traffic than other links, creating hot spots); other sequences lead to more balanced traffic. The two techniques described below try to reduce the number of hot-spot-creating destination vectors.

The first technique is statistical in nature. It is illustrated in Figure 12.21 for the banyan switch. An additional banyan switch is added in front. It is called the randomization stage. When a cell arrives at an input port, its destination output port,  $Out$ , is replaced by another destination port,  $Ran$ . The new destination port is randomly selected with a uniform distribution over all  $N$  output ports. On leaving the randomization stage, each packet’s original destination  $Out$  is restored, and it enters the second switch at input port  $Ran$ .

Thus, in the first stage, the probability distribution of the sequence of destination vectors is independent from one time slot to the next and uniformly



12.21

FIGURE

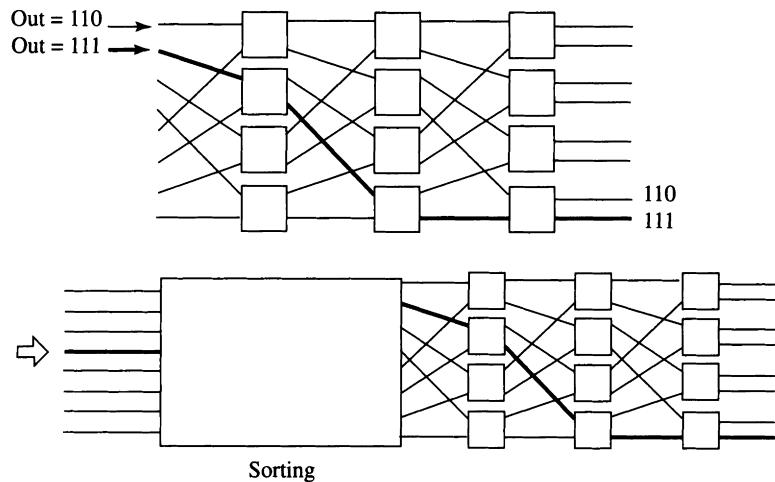
Introduction of a randomization stage balances the traffic entering the switching stage and reduces queuing delay.

distributed over the output ports. From the symmetry of the banyan (and other) switches, one can see that as a result the links inside the switches will be required, on average, to carry an equal number of packets. That is, on average, the traffic will be balanced, and the probability of hot spots will be low (in the first stage).

The randomization achieved in the second stage is a mirror image of the randomization in the first stage. Here, for the destination output port for any cell, the input port from which that cell originates is independent from one interval to the next and uniformly distributed over the  $N$  input ports. The symmetry of the switch ensures once again that, on average, the traffic over the links in the second stage will also be balanced, thereby reducing the occurrence of hot spots. Since randomization is statistical, this technique does not eliminate all hot spots, and so each  $2 \times 2$  module must contain buffers.

The second technique is more radical: it completely eliminates contention (and hence the need for buffers) within the switch fabric. The technique is based on the observation that if the destination  $N$ -vector is sorted (in increasing order) by output port, the corresponding  $N$  routes will be disjoint. This is illustrated in Figure 12.22. Cells in the first two input ports have destinations 110 and 111, so they are sorted; the other input ports are idle (idle is highest in the ordering). We can see that the two routes have no links in common, so both cells can be transmitted simultaneously, and there will be no contention or need for buffers.

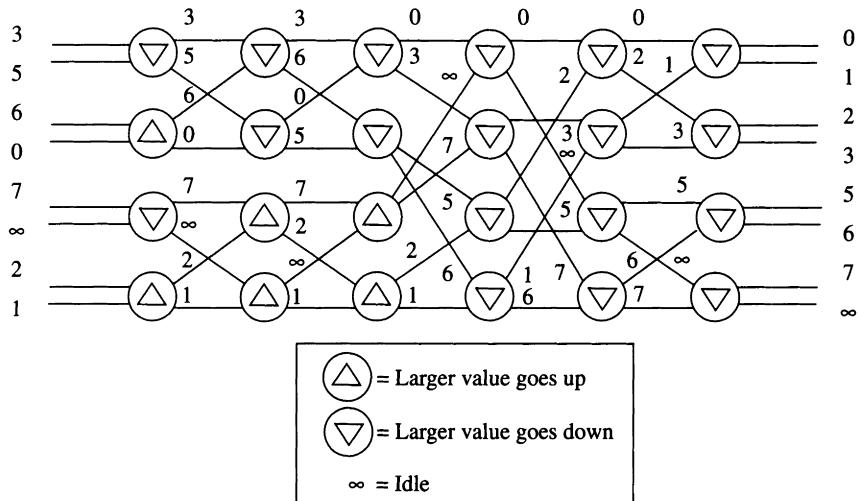
One can take advantage of this observation by a two-stage switch design in which the first stage is an automatic “sorter” in which the cells appear at the output of that stage properly sorted. Figure 12.23 illustrates one sorter called the *Batcher network*. If a Batcher network is followed by, say, a banyan switch,



12.22

FIGURE

If the destinations of arriving cells are sorted by output port, the corresponding routes are disjoint and there is no contention.



12.23

FIGURE

The Batcher network sorts incoming cells by order of output port.

the destination vector appearing at the banyan switch will be sorted, and there will be no contention inside the switch. Thus there is no need for buffers in the banyan switch.

Of course, if two input cells with the *same* destination output port appear at the Batcher network, there will be contention. So one of those cells cannot be allowed to enter the network and must be buffered. Thus a Batcher-banyan switch must be equipped with buffers at the input. Such switches are sometimes called *input buffered*. Input buffering can reduce utilization, hence throughput. For example, suppose (referring to Figure 12.22 again) that the first two input ports have the same destination port 110. Then one of the cells, say the one at the second port, must be buffered, and so this port (and the corresponding route) is kept idle for one cell time. We will study input buffered switches in section 12.8.

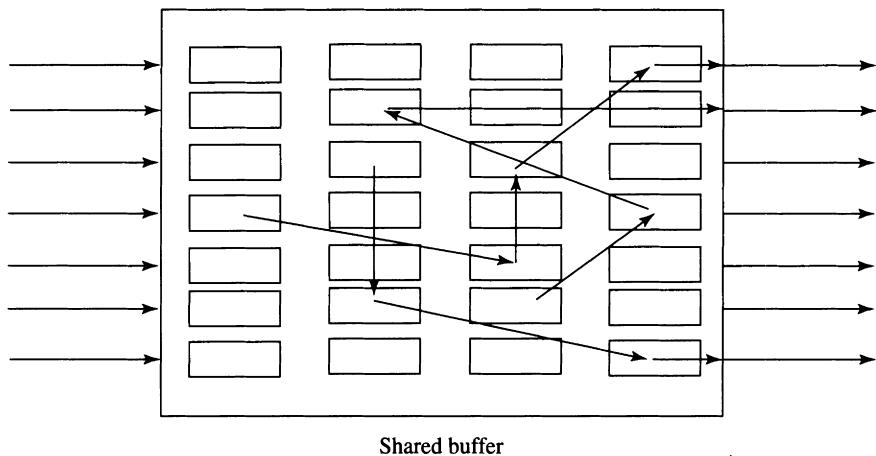
#### 12.5.4 Multicasting

We have seen that the self-routing property of distributed buffer switches makes the internal switching decisions simple. We examine how the same property can be used to implement multicasting. First, consider a cell that arrives at the omega switch and that must be sent to two output ports  $ABC$  and  $A'B'C'$ . One solution is to duplicate the cell at the input and to send one copy to  $ABC$  and the other copy to  $A'B'C'$ . We can improve the efficiency of the internal links somewhat by adding the combined tag  $[CBA, C'B'A']$  to the cell. If  $A = A'$ , then we send the cell with the tag  $[CB, C'B']$  to the output link of the first module specified by  $A$ . If  $A$  is not the same as  $A'$ , then we need to send one copy of the cell with tag  $[CB]$  to the output link specified by  $A$  and the other copy to the other link with the tag  $[C'B']$ . We repeat the procedure at the next stage. The same method can be extended to an arbitrary number of destinations and stages. Thus, if the modules can duplicate cells and manipulate tags as explained above, we are able to avoid sending more than one copy of each cell on any given link of the switch, thus making multicasting efficient.

## 12.6

## SHARED BUFFER

The modular constructions of the delta networks of Figure 12.13 are based on circuit switch designs. The other three designs are based on the design of routers for packet networks. They are not modular. In this section we study the shared buffer switch (SBS).



12.24

FIGURE

The shared buffer is divided into linked lists corresponding to output ports.

The name *shared buffer* is clear from Figure 12.24. There is a common pool of buffers divided into linked lists indexed by the output port name. (The figure shows three linked lists corresponding to three output links.) There is also one linked list of free buffers. (Thus there are  $N + 1$  linked lists in all, if there are  $N$  output ports.) Each list has a begin and end pointer. The switch operates as follows. We assume that time is slotted, with one slot per cell transmission time. In each time slot the following operations take place. First, a cell from the beginning of each linked list is transmitted over the corresponding output port. The list's begin pointer is updated, and the buffer is added to the free buffer list. Second, the cells that arrived at the input ports during the time slot are examined. If a cell is destined for output port  $i$ , it is put in a free buffer (whose begin pointer is moved up) and that buffer is appended to the list for port  $i$ .

The main advantage of this scheme over the other switch designs is that a much smaller buffer is enough to achieve the same blocking probability. In a distributed buffer switch, the buffers allocated to different modules are not shared; in an output buffered switch, the buffers allocated to different output ports are not shared; and in an input buffer switch, the buffers allocated to different input ports are not shared. As a result some buffers may be full, which may lead to blocked cells, while other buffers are not full. Another advantage is that more complex buffer-sharing schemes, involving priorities for example, may be implemented. Priorities can also be readily implemented in the input and output buffer switches, but again, at a cost of even more buffers.

The disadvantages of the shared buffer design all stem from the fact that the shared buffers must be accessed at a speed equal to the sum of the input (or output) speeds. In each time slot, one cell per input port has to be read into the buffer, and one cell per output port has to be read out. If buffers are implemented in RAM, then their speed will place a limit on the maximum throughput of an input buffered switch.

### 12.6.1 Multicasting

Consider a cell that arrives at the shared buffer switch and that must be sent to a number of output ports. That cell must be added to the linked lists of the different output ports.

Consequently, it appears that there is no efficient multicasting strategy for shared buffer switches.

### 12.6.2 Queuing Analysis

We will calculate the size of a linked list for a particular output port, port 1, say. Let  $X_t$  be the size of this list at the beginning of the  $t$ th time slot. ( $X_t$  is a random variable.) Suppose that  $A_t$  cells with destination port 1 arrive during this slot. (Thus  $X_t \geq 0$ , and  $0 \leq A_t \leq N$ , if there are  $N$  input ports.) Then,

$$X_{t+1} = (X_t - 1)^+ + A_t = X_t + A_t - 1(X_t > 0), \quad n \geq 0, \quad (12.3)$$

where we use the notation that for any number  $z$ ,  $z^+ = \max\{z, 0\}$ ; and  $1(\cdot)$  is the indicator function, so  $1(z > 0) = 1$  if  $z > 0$ , and 0 otherwise. The term  $(X_t - 1)^+$  accounts for the fact that if  $X_t > 0$ , then one cell will be transmitted out of port 1, leaving a list of length  $(X_t - 1)$ . Suppose that the arrivals  $A_t$  are iid. Assume that we have reached steady state, and let  $P(X)$  denote the steady-state distribution of  $X_t$ . Let  $P(A)$  denote the distribution of  $A_t$ . Taking expectations on both sides of (12.3) we get

$$E(X) = E(X) + E(A) - P(X > 0),$$

so

$$P(X > 0) = E(A) =: \rho.$$

Next we square both sides of (12.3) and take expectations to get

$$E(A^2) + \rho + 2\rho E(X) - 2E(X) - 2\rho^2 = 0.$$

Suppose the variance of  $A_t$  is  $\sigma^2$ , so  $E(A^2) = \rho^2 + \sigma^2$ . Then, the average size of the list is

$$E(X) = \frac{\rho + \sigma^2 - \rho^2}{2(1 - \rho)}. \quad (12.4)$$

In the special case that  $A_n$  has a Poisson distribution,  $\sigma^2 = \rho$ , so substituting in (12.4) we get the well-known formula

$$E(X) = \frac{2\rho - \rho^2}{2(1 - \rho)}. \quad (12.5)$$

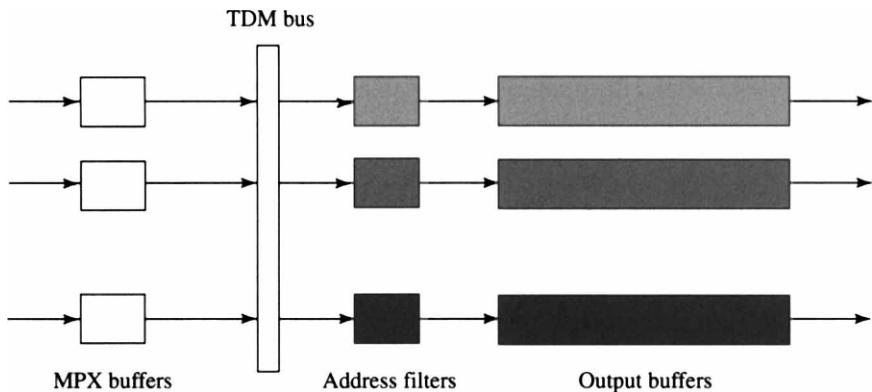
Since on average  $\rho$  cells arrive per slot destined for output port 1, and since one cell can be transmitted from that port per slot,  $\rho$  is also the average utilization of the transmission bandwidth of that output link. From (12.5) we see that if the utilization is 80%, then the average number of cells in the buffer is 2.4, and if the utilization is 90%, it is 4.95.

## 12.7 OUTPUT BUFFER

In Figure 12.25 we exhibit an output buffer switch design. (Many routers are based on this design.) Incoming cells are multiplexed over a fast bus or any other broadcast medium. (In order to facilitate this multiplexing operation, a small buffer is needed at each input port.) Corresponding to each output link there is an address filter that reads the bus, selects the cells destined for that output link, and copies those cells into the corresponding output buffer. This design gives a better buffer utilization than the distributed buffer design but a lower buffer utilization than the shared buffer design. The disadvantage is that it needs a high-speed bus. The bus speed must be as large as the sum of the input link speeds.

### 12.7.1 Multicasting

Of all four designs considered here, the output buffer design is best suited for multicast traffic. Indeed, cells with multiple destination ports can be handled with virtually no additional complexity.



12.25

FIGURE

Output buffered switch: each cell is broadcast, and a filter copies it into the proper buffer.

## 12.7.2 Knockout

A design difficulty arises with output buffer switches. The number of cells that want to enter a given output buffer in one cell time can be as large as the number of input lines, say  $N$ . Accordingly, the number of input lines will be limited by the speed of the electronics used for the output buffers. To avoid this limitation, designers propose to limit the number of cells that can be transferred into an output buffer to some value  $K < N$ . If  $M > K$  out of  $N$  cells are destined to the same output buffer in one cell time, then  $K$  of them are selected for transfer and the others are dropped. Various procedures exist for making sure that each of the  $M$  cells has the same probability of being selected, for fairness. One such scheme—called the *knockout switch*—implements a  $K$ -stage knockout tournament.

The designer must choose the value of  $K$  that keeps the fraction of dropped cells to an acceptable small value. To determine such a value, assume that the input cells have independent destinations, picked uniformly among the  $N$  output links. Assume also that there is a cell arriving at each input link with probability  $\rho$ , independently of the other input links. The probability that  $M$  cells arrive in one time slot and are destined to a specific output link is then equal to  $P(M)$ , where

$$P(M) = \binom{N}{M} \left(\frac{\rho}{N}\right)^M \left(1 - \frac{\rho}{N}\right)^{N-M},$$

where the term  $\rho/N$  is the probability that a cell arrives at a given input link and is destined for that specific output link. For that specific output link, the expected number of cells dropped in one time slot is therefore equal to  $\alpha$ , where

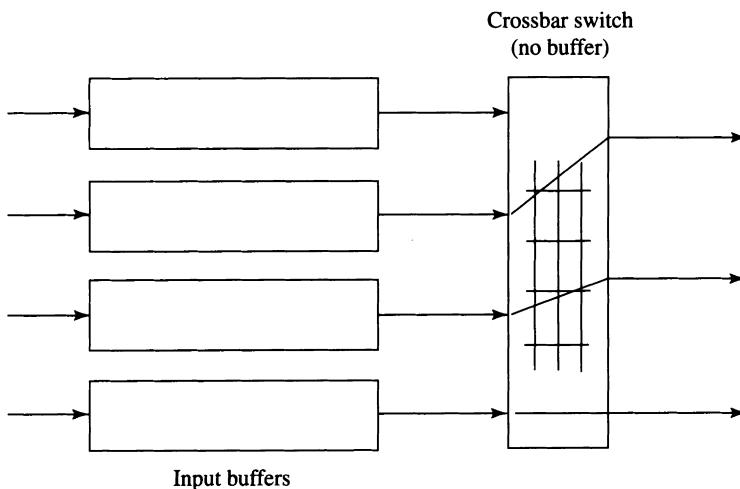
$$\alpha = \sum_{M=K+1}^N (M - K)P(M).$$

Indeed,  $M - K$  cells are dropped with probability  $P(M)$  for  $M = K + 1, \dots, N$ . Since the average rate of cells destined for that output link is equal to  $\rho$ , we conclude that the average fraction of cells dropped per output link is equal to  $\phi := \alpha/\rho$ . Using these formulas, we can determine the value of  $K$  that guarantees—under our assumptions—that the fraction  $\phi$  of cells that are dropped is acceptable. The upshot of the analysis is that the loss rate is less than  $10^{-10}$  for  $K = 12$  provided that  $\rho \leq 0.9$ , for any value of  $N$ .

## 12.8

## INPUT BUFFER

The input buffer switch design is illustrated in Figure 12.26. Incoming cells are stored in buffers, one per input port. In each time slot, the crossbar (or any strictly nonblocking) switch transfers the cells at the head of each input buffer



12.26

FIGURE

Input buffered switch: cells contend for transfer through the crossbar switch.

to their destination output ports, provided there is no contention. That is, if cells at the head of more than one input buffer have the same destination, the switch controller selects only one of these for transfer by the crossbar. (The design of this controller is straightforward if it knows the destinations of all the input cells. A more interesting problem is to design a switch controller that resolves contention in a decentralized manner.)

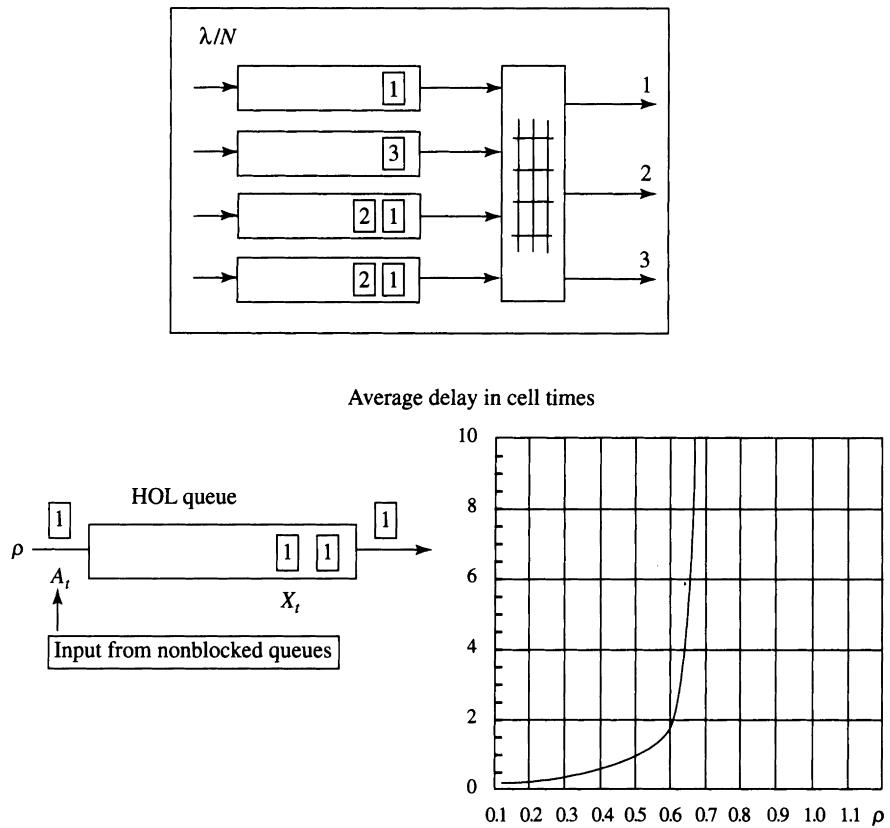
The speed of the crossbar is the same as that of the input lines. This is the major advantage over the shared buffer and output buffer designs (whose buffer and bus speeds, respectively, must equal the sum of the input line speeds). As a result, for a given very large-scale integration (VLSI) technology, the input buffer switch can have much greater throughput than these two designs. The input buffer switch, however, suffers from head of line (HOL) blocking, which can reduce its throughput considerably. We will study this next and then examine ways to overcome it.

### 12.8.1 HOL Blocking

Figure 12.27 shows a switch with four input and three output ports. Suppose that the destination of cells at the head of the four input buffers are as shown. Then, in the next time slot, the crossbar will transfer two cells: one of the cells from buffers 1, 3, or 4 to output port 1, and the cell from buffer 2 to output port 3. The crossbar will not transfer cells in buffers 3 and 4 with destination port 2 because they encounter HOL blocking. Note that the crossbar has the capacity to transfer up to three cells per slot, so that HOL blocking reduces the utilization of this switch. We now calculate this reduction.

We consider an  $N \times N$  switch with large  $N$ , and we focus attention on type 1 cells—those with destination port 1. In each time slot, a type 1 cell arrives at every input port with probability  $\lambda/N$ . (See the top of the figure.) The cells in different slots and at different input ports are iid. (For example, if one cell arrives at each input port per slot, and if its destination is uniformly distributed over all  $N$  output ports, then  $\lambda = 1$ .) Since  $N$  is large, the total number of type 1 cells that arrive at the input ports during the  $t$ th time slot has a Poisson distribution with mean  $\lambda$ . (The exact distribution is binomial.) Suppose that at the beginning of the  $t$ th time slot,  $X_t$  cells of type 1 are at the head of their input buffers. We want to study the evolution of  $X_t$ , assuming steady state. (See the HOL queue in the bottom-left of the figure. This queue is not real; it is a mathematical device for us to keep track of the number of packets at the head of their queues.) Because the crossbar will forward only one of these cells to the output during the  $t$ th time slot,

$$X_{t+1} = (X_t - 1)^+ + A_t = X_t + A_t - 1(X_t > 0), \quad t \geq 0,$$



12.27

FIGURE

A queued cell prevents later cells from access to the switch fabric, causing HOL blocking.

where  $A_t$  is the number of new type 1 cells that come to the head of the input buffers during this slot. Assume that in steady state, the probability that a buffer is unblocked is  $\Phi$ . Then  $A_t$  has a Poisson distribution with mean  $\rho := \lambda \times \Phi$ . The same calculation that led to (12.5) shows that

$$P(X_t > 0) = \rho, \text{ and } E(X_t) = \frac{2\rho - \rho^2}{2(1 - \rho)} =: \beta.$$

By definition, the expected number of blocked buffers is  $N(1 - \Phi)$ . If we assume that the destination of the cells is uniformly distributed over all  $N$  output ports, then the expected number of blocked buffers also equals  $NE(X_t - 1)^+ = N(\beta - \rho)$ . So

$$\Phi = \frac{\rho}{\lambda} = 1 - \beta + \rho.$$

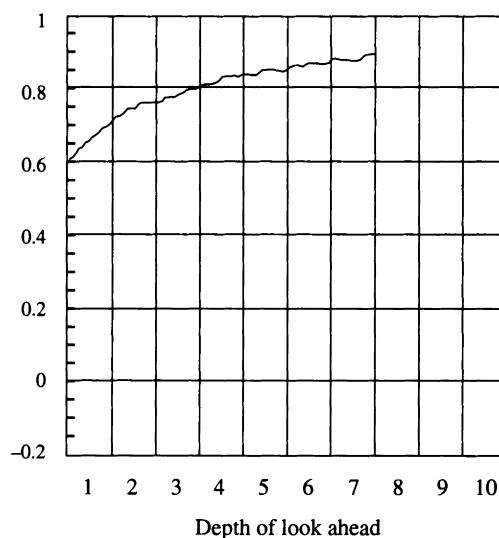
For  $\lambda = 1$ , this gives  $\beta = 1$ , or  $\rho = 2 - \sqrt{2} = 0.58$ . Note that  $\rho$  is the throughput of type 1 cells. Thus, under the assumption of iid arrivals and uniformly distributed destinations, HOL blocking reduces throughput to 58%.

Because of this, the input rate  $\lambda$  must be less than 0.58; otherwise the queues in the input buffers will become unbounded. By a more detailed analysis of the HOL queue in the left of the figure, we can calculate the average delay through the switch. As expected, the figure shows that the delay becomes unbounded as the input rate approaches 0.58.

### 12.8.2 Overcoming HOL Blocking

We now consider two techniques to overcome the 58% throughput limit. The techniques are quite intuitive. The first technique allows the crossbar to look ahead into each input buffer for cells that could be transferred if they were not blocked by the head cell. Clearly, the throughput increases with the depth of the look ahead, as shown in Figure 12.28. An infinitely deep look ahead

Throughput (simulation)



12.28

FIGURE

Throughput of an input buffer switch increases with the depth of look ahead in each input queue.

corresponds to input expansion, in which each input port has one buffer per output port. HOL blocking is completely overcome, and the utilization is 1, when one cell arrives at each input port per slot with a destination that is uniformly distributed over all output ports. (The plot in the figure approaches 1 as the depth of look ahead approaches infinity.) We analyze the input expansion case.

Fix an output port (port 1, say) and let  $X_t$  be the number of all type 1 cells waiting in the input buffers during slot  $t$ . Because there is no HOL blocking, one of these cells is forwarded through the switch in each time slot. So once again we have the equation

$$X_{t+1} = (X_t - 1)^+ + A_t = X_t + A_t - 1(X_t > 0), \quad t \geq 0,$$

where  $A_t$  is the number of new type 1 cells that arrive during slot  $t$ . If the  $A_t$  are iid random variables with mean  $\rho$  and variance  $\sigma^2$ , then (see (12.4)) in steady state,

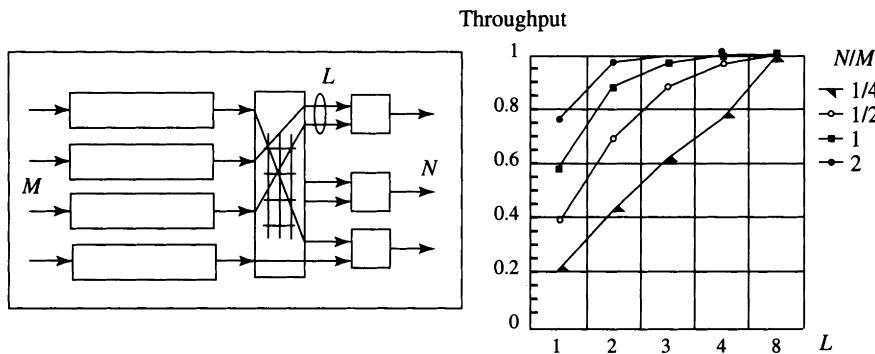
$$E(X_t) = \frac{\rho + \sigma^2 - \rho^2}{2(1 - \rho)}.$$

Let us observe how this analysis applies to the output buffer switch as well. Fix an output port (port 1), and let  $X_t$  be the number of cells waiting in the corresponding buffer. Since one of these cells is transmitted in each time slot,  $X_t$  evolves in the same way as the equation above and so the steady-state expected value of  $X_t$  is the same. In particular, the throughput and delay of the output buffer switch and the input buffer switch with input expansion are the same. There remains an important difference, however: the input buffer switch must transfer only one cell to an output port in one slot, whereas the output buffer switch must be capable of transferring  $N$  cells if there are  $N$  input ports.

The second technique uses output expansion, in which each output port is replaced by, say,  $L$  output ports. The crossbar switch can transfer  $L$  cells to the same destination (in place of one cell), thereby increasing the throughput. This is shown in Figure 12.29.

### 12.8.3 Multicasting

The crossbar switching fabric of an input buffer switch is efficient at replicating cells. Consequently, it is tempting to attempt to design multicasting strategies. Typically, one would maintain a separate queue at each input port for cells that



12.29

Throughput of an input buffer switch increases with output expansion.

FIGURE

must be multicast. The challenge is to schedule the multicast cells to use the crossbar efficiently.

To illustrate the complexity of the scheduling, assume that one cell  $C$  is waiting to be sent to outputs  $\{1, 2, 3\}$  while another cell  $D$  is waiting to be sent to outputs  $\{3, 4\}$ . It appears reasonable to send cell  $C$  to  $\{1, 2, 3\}$  and cell  $D$  to  $\{4\}$ , keeping cell  $D$  in the input queue together with the remaining destination  $\{3\}$ . Indeed, this choice maximizes the number of copies that the crossbar sends in one step. However, if cell  $D$  is followed in the input buffer by many more cells than cell  $C$  is, then the decision to complete the transmission of cell  $C$  instead of that of cell  $D$  may prove suboptimal. This example illustrates again the trade-off between short-term and long-term benefits.

## 12.9

## SUMMARY

Advances in high-speed transmission over optical fiber discussed in Chapter 11 and fast cell switches discussed in this chapter are the two technologies that make high-performance networks affordable.

There are four basic switching architectures: input buffer, output buffer, shared buffer, and distributed buffer. Switch performance is measured in terms of throughput, delay, and complexity. Switches that implement the first three architectures are limited in their total throughput by the maximum speed of electronic circuits; hence those switches are used mostly for local ATM networks. Each architecture has its own bottlenecks that cause queuing delays. Simple models are available to analyze those queuing delays. In most cases,

additional features can be added to the basic architecture to overcome bottlenecks. For example, head of line blocking of the input buffer switch can be reduced by “look-ahead” schemes.

Distributed buffer switches are modular and can be scaled to arbitrarily high total throughput, and so they are used for large wide area ATM switches. The construction borrows ideas from modular telephone circuit switches. Distributed buffer switches can suffer large queuing delays that can be overcome by a randomization or sorting stage that precedes the switch.

All switches process cells within the same connection in the same order in which they arrive (first come, first served, or FCFS service), as is required for virtual circuit connections. These switches extend FCFS service to cells in different connections that share the same input and output ports, which is not required. Indeed, in order to provide different QoS to different connections, FCFS service may be inappropriate. For instance, connections that have a guaranteed delay requirement may need to be served with a higher priority than connections that do not have such guarantees. Similarly, connections that require greater bandwidth should be served more frequently than those that require less bandwidth.

In order to provide different QoS to different connections, these switches must be enhanced. The enhancement typically consists of buffer management facilities that allow the switch to keep track of cells from different connections and to process cells from those connections in ways that meet their QoS requirements. For example, suppose there are two different priorities. Then two different buffers would be created, and higher priority buffers would be processed before the lower priority buffers. As another example, suppose delay guarantees were met by providing deadlines to cells. Then the switch would implement a buffer management policy that processes cells in order of their deadline. (Such a service is called “earliest deadline first.”)

A fast switch with these buffer management facilities would be able to provide the full range of QoS for ATM connections.

## 12.10

## NOTES

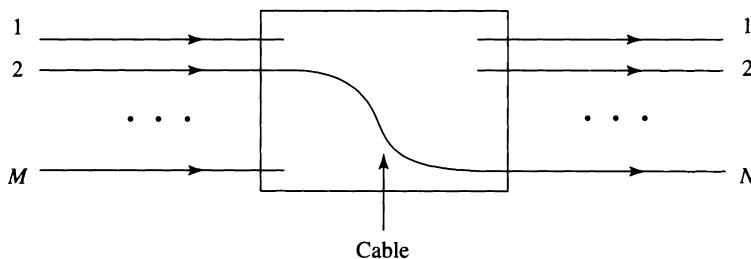
Stochastic models for packet switching and traffic are developed in [H90]. A collection of papers on high-speed switching is available in [R93]. A general discussion on queuing models in data networks is available in [BG92]. A recent approach to the analysis and control of high-speed packet networks is proposed in [FMM91]. The distributed buffer switch was invented as a means of

connecting multiple processors to multiple memories. The unbuffered switch is modeled and analyzed in [P81, KS83], which also estimate the hardware complexity. The buffered case is analyzed in [J83, KL90, YLL90]. The input buffer switch is analyzed in [KHM87, HA87]. The benefit of speeding up an input buffer switch is evaluated in [OMKM89]; look-ahead schemes to overcome HOL blocking are studied in [L90, MVW93]; the effect of output expansion is investigated in [LL91]. The knockout switch is presented in [YHA87].

## 12.11

## PROBLEMS

1. A circuit switch has  $M$  input ports and  $N$  output ports as in Figure 12.1. The switch is controlled by an operator who has one cable that can be used to connect any one input port to any one output port as in Figure 12.30. How many switch configurations can the operator reach? Design an arrangement with crosspoints that achieves the same configurations. Repeat the question for the case that the operator has  $k$  cables.
2. Determine the tag-to-route encoding for the baseline and flip networks of Figure 12.13. Assume as in Figure 12.16 that 1 denotes down and 0 denotes up.
3. Show that if two banyan switches are connected in tandem, the resulting switch is rearrangeably nonblocking. How many paths are there between any pair of input-output ports?
4. In a  $2^n \times 2^n$  DBS switch, show that there is a unique route through the switch between any pair of input-output ports. How many such routes are there in a Benes network?



12.30

FIGURE

The cable is used by the operator to connect any input port to any output port.

5. Show in the omega switch that if cells arriving at the input ports are sorted in increasing order by output port, then the routes of these cells are disjoint. What happens in the other switches?
6. In the Batcher-banyan switch there is no need for buffering in the second stage. However, if cells appear in two or more input lines destined for the same output port, then all except one of these must be buffered. Show that this network behaves like an input buffer switch. In particular, show that the switch is subject to HOL blocking.
7. How would you obtain the steady-state distribution of the HOL queue?
8. How would the analysis of section 12.8 be modified for the case where the cell arrivals are iid but the destination is not uniform over all output ports?
9. A shared buffer switch is built from RAM with an access speed of  $1 \mu\text{s}$ . Suppose the RAM is organized in 16-bit words and the cells are 48 bytes long. What is the throughput of this switch? If this is a  $4 \times 4$  local area ATM switch, what is the maximum line rate the switch can support?
10. Analyze the buffers saved in output buffer compared with distributed buffer, and shared buffer compared with output buffer.
11. Prove that a 2-by-2 switching module with Bernoulli arrivals is stable when the arrival rates are such that the average rates on the links of the switch are less than 1.

*Note:* A quadratic function of the queue lengths is a Lyapunov function for the Markov chain.

# Toward a Global Multimedia Network

In Chapters 3, 4, 5, and 6 we studied networks from the viewpoints of the physical layer (bit ways) and the bearer services they provide. Datagram networks provide best-effort transfer of messages, circuit-switched networks dedicate bandwidth along a route for transfer of constant bit rate streams, and ATM networks can dedicate resources for transfer of variable bit rate traffic with differentiated quality of service. In Chapter 7 we studied wireless networks, the principal means for providing “anytime, anywhere” access to telephone and data networks. In Chapters 8, 9, and 10 we studied the technical and economic means of network control. Finally, in Chapters 11 and 12 we described optical-link and fast switching technologies that make high-speed networking affordable.

In this brief chapter we take a broader perspective starting with two questions. The first question asks, what will be the distinctive features of the global, multimedia network of the next century? We address this question by discussing five necessary attributes of a global network. These attributes accommodate service quality variation, heterogeneity, mobility, extensibility, and security.

The second question asks, what advances are needed to build a network with these five attributes? We frame these advances as challenges to three technology areas: signal processing, networking, and applications. (This book is primarily concerned with networking technology. However, the two other technology areas are essential to a successfully deployed network: without successful applications there will not be enough users, and without signal-

processing advances, it will be prohibitively expensive to use the network to transfer video and images.)

We indicate some recent developments that suggest how these advances might come about.

These three sets of technologies must be used to implement an architecture that can accommodate the five attributes of the global network. We suggest requirements that this architecture must meet, pointing to the shortcomings of some of the current architectures. Since this chapter raises many questions, few of which are answered, the chapter might be regarded as a formulation of a research agenda for a global network.

In section 13.1 we explain the five attributes of the global network, and in section 13.2 we introduce the technological areas and selective recent advances. In section 13.3 we propose requirements of the global network.

---

### 13.1 ATTRIBUTES OF THE GLOBAL NETWORK

The phrase *global network* conveys the misleading image of a single unitary network. In reality, the global network will be an interoperable collection of networks that supports multimedia applications incorporating data, audio, graphics, video, images, and animation. This network will offer cost-effective, high-performance services, including entertainment-quality video. The network will be scalable to support millions of users, and flexible and extensible to accommodate future applications.

The global network will be heterogeneous in many dimensions. There will be multiple constituent networks, including the public telephone network, the Internet, extensions of the current CATV distribution systems, satellite networks, packet radio networks, and local area networks. These networks individually and collectively incorporate disparate transmission technologies, including fiber optics, wire-pair, coaxial cable, microwave, radio, and infrared wireless. There will be a variety of terminals with widely differing capabilities for display, playback, and processing, ranging from battery-operated wireless personal digital assistants or PDAs and PCs to multiprocessor supercomputers.

This heterogeneous infrastructure of networks and terminal equipment will support a wide and dynamic mix of applications, addressing the needs of small, specialized groups of users (e.g., users remotely running supercomputer applications requiring very high communications bandwidth) as well as com-

mon applications like telephony, e-mail, videoconferencing, information and entertainment delivery, and electronic commerce with millions of users.

Several major industries (software, semiconductor, computer, telecommunications, and content providers), tens of standards bodies, and hundreds of hardware and software vendors will participate in the design and deployment of this global multimedia network.

Largely unaware of the technologies and organizations involved, users will want applications to operate seamlessly across the network infrastructure, with applications and networks appropriately scaling and configuring to whatever detailed technological components are involved. Users may want their applications to be restricted to a portion of the network (and hence to a subset of the other users) or to equipment from a particular vendor. Most of the parties involved—users, service providers, content providers, and equipment vendors—will want a flexible and dynamic network that can scale and evolve to meet whatever demands are placed on it and to accommodate new, unanticipated applications without major dislocation or investment.

These considerations suggest that the global network must have the following five attributes:

- ◆ Heterogeneity—the ability to deal with a large variety of transport and terminal technologies and applications.
- ◆ Quality of service—the ability to reserve resources within the network and terminal devices so as to ensure that certain perceptual or objective performance measures are met.
- ◆ Mobility—the ability to provide a moving access point to the network.
- ◆ Extensibility—the ability to accommodate a variety of new applications and users in the future. There are two aspects to extensibility: first, all architectural features of the network must scale to an arbitrary expansion of users and applications needs; second, the architecture must be able to accommodate new technologies and applications.
- ◆ Security and reliability—including the ability to ensure that user communications are not intercepted and that their location is not tracked and also to assure the high availability of network services.

The following example illustrates some of these attributes. Audio and video compression algorithms in the global network must accomplish much more than minimizing the bit rate, as in the past. They must be capable of scaling to a variety of transport scenarios, from circuit- to packet-switched, from broadband fiber optics to wireless, from high reliability (fiber) to low reliability (mobile

wireless), and from low time jitter (constant bit rate or circuit-switched) to high jitter transmission. The compression algorithms must be able to work with a concatenation of several of these transmission scenarios, dealing, for example, with broadband backbone connections with and without wireless access.

These algorithms must also be able to scale to differing levels of processing power and resolution in the receiving terminals while providing the best subjective quality possible. They must also accommodate new types of services, such as multicast (several simultaneous receivers, as in a videoconference) and mobile users who have a changing access point to the network. In addition, these functions must be carried out in a way that is compatible with end-to-end encryption for privacy that prevents the network from "looking into" the compression syntax or processing it in any way.

All these objectives must be accommodated in an architecture that manages the inherent complexity possibly by providing the appropriate abstractions and modularity, so that the different underlying technologies can evolve separately and new technologies can be accommodated. This implies that highly configurable generic resources (such as reliable bandwidth, resolution, and delay) can be used in a negotiation between applications and networks at the time of service establishment. All this points to an elaborate set of interactions among the constituent parts of the system and the need for a carefully crafted architecture that structures those interactions in a useful manner. Mechanisms such as Sun Microsystems' Jini are being developed to enable the self-organization of devices into a network. In such mechanisms, the devices can signal their existence and characteristics as well as discover the other reachable devices. For instance, a printer may advertise itself as it is brought on-line, and the networked computers may then select the suitable driver to be able to use it.

---

## 13.2 TECHNOLOGY AREAS

Three sets of technologies are used to build the global network: networking, signal processing, and applications. These technologies are embodied in elements that must be organized within an appropriate architecture in order to provide the attributes listed in the previous section.

Networking technologies are concerned with providing quality end-to-end transport between service providers and consumers. Networking technologies must be able to transport diverse multimedia data types across heterogeneous link technologies and independent subnetworks while supporting user mobility and protecting user privacy. Networking technologies include transmission,

multiplexing, switching and routing, communication protocols, administration (billing and security), and network management.

Signal processing is concerned with the encoding, compression, storage, and playback of video, audio, and image signals. Signal-processing technologies include algorithms for coding and compression, encryption, and error correction and implementations of those algorithms in hardware or software. The algorithms must be well matched to both the transport capabilities of the network and the multimedia demands of the applications.

Applications technologies are concerned with determining and developing generic applications that will ease the development of user applications. The File Transfer Protocol (FTP) is an example of a generic application. A large fraction of future applications will incorporate multimedia (data, graphics, audio, video, and animation), as illustrated by the success of the CD-ROM and the World Wide Web. The development of HTML (Hypertext Markup Language) and associated browsers provides another example of a generic application that has popularized the use of Web pages.

A third example is offered by MBone. MBone is an IP application that creates connections between one user (speaker) and a group of other users (subscribers); new subscribers may join or leave. A naive way to implement MBone would be to create one separate connection between the speaker and each subscriber. If a connection uses up a bandwidth of  $B$  bps and the speaker's link has a capacity of  $NB$  bps, then there could be at most  $N$  subscribers. As the number of users attempting to subscribe approaches  $N$ , the speaker's node would be unable to cope, and service to each subscriber would deteriorate. Thus the naive approach does not scale.

Instead of the naive approach, MBone creates a spanning tree of subscribers rooted at the speaker: a new subscriber is attached to the "nearest" connected subscriber, who forwards a copy of the speaker's packets to the new subscriber. The copy function is now spread over many subscribers, and there is no limit to the number of subscribers that can be connected. The MBone example has many interesting features. First, the approach scales. Second, the idea can be extended to other functions. For instance, frequently accessed information from a particular Web site could be stored at intermediate nodes, reducing the burden on the source Web site. Third, MBone service can be introduced incrementally, being available only from those routers that implement the associated spanning tree and routing protocols.

### 13.2.1

### Architecture

The network architecture provides a framework for organizing the functional elements needed for the global network. The elements must be modular (that

is, specified independently of each other) so that different implementations can realize those elements in ways that encourage the use and development of technological innovations. The modularity of the Internet and OSI architectures has permitted the immediate incorporation into networks of higher-speed computers, links, and switches.

However, as we have often stressed, many applications, particularly multimedia applications, demand dedicated resources at the physical layer. Meeting such demands may create a dependence between the application layer and the physical layer of the architecture. The architecture must be carefully designed so that such dependence does not compromise modularity. An equally important requirement on the architecture is that it must accommodate existing networks.

We are concerned primarily with technological challenges to global networking, but economic and social challenges must be overcome as well. Access to the global network must be sufficiently cheap if it is going to be global. (In many developing countries today, large numbers of people do not have access to telephone service.) And the global network must be deployed in ways that contribute to social progress.

### 13.2.2 Networking

The current networking strategies (multiplexing, switching, flow control, and congestion control) of the Internet are remarkably successful for applications that require only best-effort transmissions. These strategies must evolve to provide the stringent QoS required by multimedia applications. The QoS must be estimated for network services, and a signaling procedure must be defined to enable the selection of the QoS across different networks. The strategies must take signal processing into account for several reasons:

- ◆ The subjective fidelity at the application layer depends on the compression, encryption, and temporal characteristics of the losses at the buffers. (For example, are losses independent, positively or negatively correlated?) Currently, such temporal characteristics are not known, and methods for controlling them are not available.
- ◆ The losses depend on the statistics of the traffic and the buffer and bandwidth allocation strategies (as we have seen in Chapters 8 and 9).
- ◆ The statistics of the traffic depend on the compression, encryption, traffic shaping, flow control, buffer, and bandwidth allocation, and on adaptive/predictive routing for mobile applications.

- ◆ If the application is carried by a number of streams (e.g., one stream for audio and another for video), these streams may have different traffic statistics and QoS requirements.
- ◆ The QoS provided to mobile users may specify that the resolution of the video stream can drop when a user leaves the building.
- ◆ The security of an application may prescribe different levels of protection for distinct subsets of users.

For multimedia networking, wireless access techniques must provide high bit rates and low latency, often while accommodating a high density of users. Facing limited transmission bandwidth, available approaches seek to increase multiuser system transmission capacity (the product of bit rate and number of users per unit area or volume) by implementing sophisticated signal-processing techniques, such as joint detection and interference cancellation. However, the high complexity of these techniques will increase latency.

### 13.2.3 Signal Processing

The main standards in video processing are H.261, MPEG1, and MPEG2. These are well suited for storage and continuous bit rate transmission in the absence of errors, but they are less than ideal for networking, for these reasons:

- ◆ They are subject to propagation of errors over many frames, which impacts the QoS in case of losses, delays, or transmission errors.
- ◆ They do not permit multiresolution representation of video, which reduces their usefulness in heterogeneous environments.
- ◆ They cannot adapt to changing network conditions (congestion, variable throughput), which again limits the QoS.

The encoding of information currently enforces a separation of source and channel coding. This separation limits the performance when applications share wireless channels.

Encryption and compression accumulate their error-propagation effects. Such error propagation could be controlled by an integrated approach that develops compression and coding algorithms that match the network environment.

### 13.2.4

### Applications

Current applications are often designed assuming extreme properties of the service offered by the network. Thus, a client/server application may be designed to work properly even when packets are lost, replicated, or misordered. For multimedia applications that often involve QoS and real-time constraints, it will be impossible to design those applications to provide satisfactory performance despite such worst-case assumptions about network service. Instead, the application must be designed under the assumption that the host's operating system running the application knows (for instance) the delay and loss characteristics of the network and schedules operations accordingly. But adapting applications to the network in this way couples the operating system to the network. Such dependence of the application on the network is a major departure from current practice. Here are examples of such dependence:

- ◆ A video connection may still be satisfactory if the picture occasionally switches to black and white.
- ◆ A mobile application must be designed to handle variations in the delay and loss rate, possibly through fail-soft solutions.
- ◆ The selection of servers should take transmission delays into account.
- ◆ The setup of multimedia connections may involve resource reservations that require information about pricing and network characteristics (as opposed to RSVP, where the routing is decided beforehand).
- ◆ The selection of parameters of connections (e.g., leaky-bucket parameters or other traffic-shaping options) requires estimating the impact of a given selection on the subjective quality of the application.

---

## 13.3

## CHALLENGES

Historically, the disciplines of signal processing, networking, and applications have evolved with little interaction. As explained above, we believe that to address successfully the challenges of multimedia networking, researchers must collaborate across these disciplines and focus on the ultimate design goals. This need for a synergy across research disciplines can be organized in six areas: (1) architecture, (2) QoS, (3) mobility, (4) heterogeneity, (5) extensibility, and (6) security.

### 13.3.1 Architecture

A multimedia network incorporates many different technologies, as well as myriad functional and applications requirements. Unfortunately, these interact in complex ways, creating undesirable dependencies. Here are some examples:

- ◆ Joint source/channel coding (coordination of channel impairments with source characteristics and subjective effects) is necessary for high-traffic capacity on wireless access links, where high capacity is important. However, this creates an unfortunate dependence between the design of channel coders (at the physical/link layer) and source coders (audio and video compression) that runs directly counter to the desire for generic networks running generic and flexible applications.
- ◆ Encryption (for privacy and security) hides the basic syntactical and semantic components of a media stream, precluding any operations that “look into” or “process” that stream, such as bridging for multicast connections, conversions from one resolution to another, or conversions from one protocol or compression standard to another.
- ◆ Delay is lower-bounded by propagation delay, which is already appreciable for interactive applications on a global scale (on the order of several hundred milliseconds). The architecture must be quite disciplined to avoid adding appreciable processing or queuing delays, an issue that has been largely ignored for applications initially deployed in local area and national networks. This affects decisions about conversions from one protocol or compression standard to another, packet sizes, network topologies, and so on.

Research on architecture definition attempts to identify functional interdependencies, quantifies the relationship between overall performance parameters and architectural choices, and defines architectural concepts that satisfy user and application needs. One large challenge is to define the appropriate modularity and system abstractions so as to maintain the greatest possible independence in the design of the different parts of the network and at the same time meet performance, cost, and functionality objectives. As mentioned before, the objective of the research is not the definition of a totally new infrastructure; rather, the objective is to incorporate into the future multimedia networking infrastructure existing component networks (such as the Internet,

public telephone network, CATV networks, and future extensions) with minimal modification (this is one aspect of the heterogeneity issue discussed later). What are the minimal necessary changes?

### 13.3.2 Quality of Service

Quality of service refers to network performance measures such as delay, loss, and corruption, as seen by the users and applications. (*Corruption* is the reduction in the quality of the information perceived by the user because of quantization, compression, and loss.) A traditional approach is for the application to place constraints on the offered traffic and the network to provide QoS guarantees. While many data-oriented applications can live with best-effort networking (with no QoS guarantees, other than reliable delivery), audio and video generally require QoS guarantees. Delay must be upper-bounded to ensure real-time delivery, and loss and corruption must be bounded to ensure a predictable subjective quality. The wide range of requirements, from those with relaxed to stringent QoS parameters, suggests that it would be highly advantageous (mostly in terms of carried traffic) for applications to have control over the QoS. This is especially critical in wireless access, where providing all services a reliability high enough for the most stringent service (like MPEG video) would be prohibitively expensive. A few of the many research issues raised by QoS are described next.

A major challenge is defining appropriate measures of QoS that strike the right balance. From the application view, they should capture the essence of impairments with a detail sufficient to predict subjective quality to the user. From the network view, they need the simplicity to enable monitoring and control mechanisms for guaranteeing QoS. These measures should (in contrast to current approaches) capture temporal behavior (time correlations), and they must support features such as aggregation and disaggregation, explained on the next page. Various other dependencies have to be accommodated. The subjective quality of audio and video presentations depends on the compression and encryption algorithms used and their error-propagation effects and the temporal characteristics of the losses and corruption. The temporal characteristics of losses depend on the aggregate source traffic statistics. The traffic statistics depend on the compression, encryption, traffic shaping, flow control, buffer, and bandwidth allocation, as well as adaptive/predictive routing for mobile applications. To date, only oversimplified measures of QoS, security, and reliability have been defined, such as the delay-throughput characteristic, cell delay variation, cell loss rates, security of point-to-point links, and the probability of failure of unreliable links.

The first step in controlling QoS is to be able to predict it (using analytical and numerical tools) on the basis of resource allocations within the network. One point that is too often overlooked in the evaluation of QoS is the distinction between open-loop and closed-loop applications. An open-loop application is one that generates a traffic stream that is not affected by the state of the network. Examples include voice over IP (VoIP) and stream applications such as Real-Video and Real-Audio. By contrast, the traffic that closed-loop applications generate is affected by the network congestion. Applications built on top of TCP, such as TELNET, ftp, and HTTP, are closed-loop. For closed-loop applications, which in 1999 constitute approximately 85% of the Internet traffic, there is no notion of a “traffic model.” For such applications, an approximate model may specify random times when users request files and the random sizes of the files. Note that such a model may prove too simplistic. Indeed, the activity of a user depends on the response times of the network, so that the request times and sizes are not really specified ahead of time.

In 1999, open-loop applications make up only a small fraction of the Internet traffic. However, these applications are QoS-sensitive. For such applications, in addition to the models described in previous chapters, new traffic models are needed in broadband networks, possibly including self-similar models and multiple-time-scale models (see references at the end of this chapter). The next step is to define control algorithms to complement stochastic dynamic programming. As an example, we mention a game-theoretic model for the study of resource-allocation problems based on the observation that regulated traffic within the network is bounded by an affine function of the time interval, as is the case with traffic regulated by GCRA or leaky bucket (see section 9.4.1). Methods for predicting and controlling QoS may be based on statistics collected in a systematic way from actual network implementations (see section 8.4.3). On-line estimation methods and associated adaptive control strategies will be essential. Related work is required for routing, fault detection, adaptive compression, and other control functions.

One key concern is to understand the *aggregation* of QoS, that is, the prediction of QoS based on multiple sources of impairment, such as in tandem transport links. For example, an aggregation approach based on affine bounds turns specified bounds on the traffic into bounds on end-to-end delays. (ATM Forum recommendations based on GCRA follow this approach.) The product-form result for average delays in datagram networks discussed in section 8.3.3 is another example. Based on effective bandwidth theory, decoupling techniques for aggregating the performance measures of fast packet switches provide a third example. It is necessary to extend these approaches or to develop new

approaches to aggregation, especially emphasizing the heterogeneous network context where qualitatively different impairments must be aggregated.

The inverse of aggregation is the problem of *disaggregation* of QoS, that is, to determine the characteristics of network elements needed to achieve a given end-to-end QoS for the application. For instance, given the acceptable average end-to-end delay of a connection across several networks, we must allocate acceptable delays to the different networks and then to the various links of each network. This is straightforward for simple measures like worst-case delay, but further research is needed to treat more sophisticated measures that capture temporal behavior and statistical fluctuation. Both aggregation and disaggregation are management plane functions, together with admission control, routing, and network configuration. In this context, the design of scalable QoS strategies is imperative. The backbone network cannot keep track of individual connections and must necessarily aggregate connections into classes. The engineering problem is the design of QoS strategies with bounded complexity in the various components of the network. For instance, can one justify the use of a few classes of service in the backbone, say with DiffServ, and a more refined QoS in parts of the network that handle fewer connections?

There are other QoS-related research issues. Unlike today's networks, which simply react to application needs, multimedia networks require a back-and-forth negotiation between application and network, based on trade-offs between price and quality measures and dependent on current traffic conditions. There are many questions relating to the implementation of this negotiation and the delay it will introduce in session establishment, especially in the face of heterogeneous networks with many service providers. One approach uses software agents that allow the negotiation to be logically distributed but physically centralized, reducing the establishment latency while allowing complex traffic-dependent negotiations.

Limited transmission bandwidth will be a major factor constraining the QoS obtained in wireless access to multimedia networks. Achieving high reliability in wireless links is also expensive in terms of traffic capacity. There is also a strong dependence between delay and reliability on wireless links. These factors imply that the ability to configure applications and specify desired QoS to the wireless link is especially critical. Servers that compress information that must be transmitted over wireless links can automatically adapt applications to the characteristics of such links. There is a need to examine technologies that hold the promise of high-bandwidth wireless links, such as wireless infrared and millimeter-wave radio. Experimental research suggests that the infrared wireless links may provide a bandwidth of 100 Mbps with a bit error rate of

$10^{-9}$ . Millimeter-wave radio can provide ample bandwidth, but it is necessary to overcome difficulties like multipath.

Maintaining QoS in the presence of failures is important in some cases. Whereas a datagram network adapts smoothly to link and node failures, such failures are typically fatal for virtual circuit connections. There are promising suggestions for network designs that are robust against failures. One approach is the hierarchical organization of virtual circuits that keeps the size of routing information in the nodes under control and that enables automatic rerouting when the network detects a failure. Another approach is redundant transmission of critical information along disjoint paths (this is implemented at the bit level in SONET rings). In some multimedia applications, the stream can be partitioned into substreams with different levels of importance. This opens the possibility for fail-soft scenarios, in which only less-critical portions of streams are lost or excessively impaired.

Although current proposals for QoS specification are based on static application requirements, this is not necessary. One alternative is adaptive control of the application (such as adaptive video compression), possibly based on QoS monitoring parameters. For video and audio, where subjective impairments predominate, it might be better to vary subjective quality measures like distortion than to have excessive time jitter. (This is in sharp contrast to how data should be treated.) For example, video compression has been based on a circuit (fixed delay and bound on loss probability) model of transport. It may prove necessary to reexamine video (and audio) compression within the context of more complex impairments characteristic of packet networks. One idea is that of asynchronous video, which reduces the perceptual delay to the user by assuming an asynchronous model for both transport and reconstruction. Similar opportunities for coordinating compression and transport, through control, configurability, and scalability, need to be explored. QoS can also be achieved by structuring the application; for instance, multicast connections allow resource sharing over different receivers.

Current video coding research is moving beyond the definition of fixed compression algorithms. Instead, one can define a toolbox of algorithms from which various compression schemes can be derived. That is, a compression language or syntax is defined, rather than a specific coder. Efficient software for this compression language then becomes a key requirement. The coding algorithm specification now becomes part of the video data, and the choice of the algorithm becomes part of connection establishment. Such an approach to video coding must resolve issues of implementation complexity on general-purpose processors, and this will influence algorithmic choices in a novel way.

The flexibility so achieved should also make adaptive coding (i.e., coding tuned to a particular video sequence) much easier. One key advance that is expected is a software-only encoder/decoder for multiresolution multicast. Such advances suggest systems where the compression/decompression code might be sent along with the video stream to program the hardware as needed. A network built with such programmable devices is called an *active network*. The idea is that network components, from routers to “middleware” servers (compression, encryption, directory, search engines, caches) can be programmed by the applications themselves. Of course, reliability and security aspects of such schemes must be studied with great care.

Pricing is an effective mechanism for allocating scarce resources, and it will be an important component in establishing QoS in future networks. Pricing, and the related billing mechanisms, raise several issues: the heterogeneity, security, mobility, and privacy aspects of these mechanisms must be studied together with the impact on the QoS. The use of incentives and disincentives to force desired behavior on the part of agents is an old technique in economic theory. The multimedia network will constitute an enormous information economy, and one of the most promising research avenues to investigate is how pricing can provide the right incentives to ensure that all applications can receive their requisite QoS.

### 13.3.3 Mobility

Many of the existing algorithms for mobile network management were developed for cellular telephone networks. On entering a new cell, the user must be authenticated, billing must be established, and the connection must be rerouted to the new end-point base station. Cellular telephone systems assume centralized tracking and control and relatively large cells, resulting in infrequent handoffs. These assumptions are no longer appropriate within a microcellular in-building network or the emerging personal communications service (PCS) networks, where handoffs could occur much more frequently, low handoff latency is needed for the seamless delivery of continuous-media streams, and the base stations are likely to be interconnected by high-speed, low-latency wired networks.

For mobile networks, handoff represents the most significant challenge, primarily because of the latencies it introduces, associated with rerouting of connections and restarting packet streams as users move through the network. While establishing the reroute itself may not entail much latency, continuous-media streams are likely to be buffered at base stations, and restarting these streams back from their source will introduce significant latency. For example,

the existing Mobile IP algorithms for mobile handoff invoke heavyweight routing and forwarding machinery characterized by interruptions in service that are measured in seconds. Such large overhead is intolerable for many latency-sensitive, media-intensive applications.

For many applications, (near) real-time audio/video and collaborative support for workers on the move is a requirement. To support users and applications roaming across such a heterogeneous collection of networks, the applications must be able to adapt to the available network performance. It may appear that the concept of multiple overlay networks will be impeded by the mobile host's need for multiple transmitter/receiver systems. But already today it is possible to simultaneously configure a laptop with network adapters for in-room diffuse IR, in-building RF, campus area packet radio (connected to the serial port), and wide area CDPD (replacing the floppy drive). New technological developments are likely to yield multimode radios that integrate such alternatives into a more convenient package in the near future. Simply extending the wireless network across these multiple overlays will not be enough. Wireless network management algorithms must be redesigned to better scale with larger numbers of users. And unlike wired networks, the wireless network must handle unpredictable increases in the densities of mobile hosts and their movement patterns.

The key to success is the development of hierarchical and distributed algorithms that can scale as the size of cells decreases, the number of cells increases, and the density of users and bits per second per user dramatically increase. The ability to track the motion of users through the network, and to exploit information about feasible and probable trajectories of such motions, will enable more localized decision making and more scalable management of network resources.

Tracking users and exploiting information about their location and the physical nature of their environment is crucial for effective network management. For example, algorithms can exploit the location of users and their physical environment to yield lower-latency handoffs as well as better allocation of network resources to high-traffic areas (see predictive routing below).

Handoff across cells can introduce significant latencies, reducing the quality of continuous-media streams. This problem may be mitigated through predictive routing. If users can be tracked by the system and their likely next cells determined, one strategy for reducing the impact of handoff is to utilize processing, buffering, and backbone-network bandwidth to selectively multicast the packet streams to the probable next base stations. Selective multicast with duplicate packet-stream buffering gives the network management algorithms more flexibility as to when to execute the handoff, since the new base station

will have already been “precharged” with the packet stream. Complexity remains, however, in ensuring that duplicate packets are correctly filtered by the mobile device and that buffered packets, no longer needed to support the handoff, are deleted with minimal overhead.

Conventional mobile handoffs are implemented within a homogeneous wireless subnet. We may call these *horizontal handoffs*. The mobile host or associated base station detects a degraded signal as it reaches the fringe area of its cell. In mobile-assisted handoff, the mobile listens for beacon signals from base stations in adjacent cells, choosing to register with the cell with the strongest signal. *Vertical handoffs* allow mobile hosts to roam between heterogeneous wireless overlays. The mobile host, or higher-level network management, determines when to switch the connections to an alternative overlay network, driven by signal quality, network load, or the costs of using one overlay versus an alternative.

When the characteristics of the transport between users change during a connection, it may be advantageous to employ adaptive connections, that is, the compression and other aspects of the system may be adapted to maintain a suitable QoS. Special buffering techniques can be used to cope with fading channels. Wireless transmitters must be flexible enough to adapt to changing channel characteristics. Source-coding algorithms have been well studied for traditional transmission media, such as telephone channels. However, the study of source-coding methods and their interaction with transmission mechanisms in time-varying and error-prone environments, such as packet networks or wireless channels, is in its infancy.

At a higher level, it may become important to develop application program interfaces (APIs) that make it possible for applications to discover from the network that their communication characteristics have changed as a result of movement through the network. Much of the existing work assumes that applications can negotiate a level of QoS, and if this cannot be obtained, the service is denied; there is no fall-back strategy. Applications like Netscape have been finely crafted to work effectively in communications environments with limited connectivity. No general bandwidth- and location-sensitive API has been proposed that would provide an application with access to a range of bandwidth-sensitive, end-to-end compression, and synchronization strategies.

Another important aspect of user mobility is the portability of the user environment. As a user moves from his office computer to his home computer, he might want to face the same desktop or at least a compatible desktop that may be adapted to specialized tasks that he performs at these different

locations. An automated configuration application could learn about the frequently performed tasks and reconcile the different desktops based on user preferences.

Connection-oriented services with performance guarantees provide useful network support for multimedia applications. The guarantees are typically achieved by reserving network resources in advance. (Current ATM Forum proposals are of this type.) The dynamic nature of mobile handoffs introduces complications; it is impractical to reserve all possible future channels. Rather than tearing down connections only to rebuild them, there may be an incremental strategy that modifies existing connections by partially reestablishing them after a horizontal handoff. Such an approach exploits the locality of logically adjacent cells to limit the amount of work involved in reestablishing the connection. Since the established channel from the packet stream source to either cell will largely be along the same route, only the “tail” of the connection needs to be rebuilt.

This illustrates a general strategy worth exploiting in mobile networks—the so-called gateway-centered approach investigated in the early days of packet radio. Connections are routed to a logical gateway for a region of the mobile network. Movement between gateways is a relatively rare event. For example, a gateway might map onto a building or a section of a campus. The gateway hides the complexity caused by local movement of mobile hosts from the rest of the network by providing local routes and performance guarantees to the mobile host within its region.

For “black pipe” networks with no performance guarantees, it is still desirable to attempt to characterize end-to-end network performance. One possible mechanism is for the network management layer to inject periodic measurement packets into the subnet to characterize the route’s latency and variability. An alternative, available in networks that expose some level of control to higher layers, is to exploit out-of-band measurements of the QoS characteristics. This makes it possible for network management algorithms to make their decisions based on more accurate and timely characterization than would be possible with periodic end-to-end measurements. These kinds of QoS measurements can best be exploited by source- and receiver-based rate control mechanisms in the mobile networking and application support layers. For example, they can be used to determine how aggressively a mobile application can pursue buffering, prefetching, and compression strategies. Wireless spectrum is not free, so look-ahead should be applied only when the application can expect a high hit rate in cached data.

### 13.3.4

### Heterogeneity

Heterogeneity will be an inescapable feature of global networks. It manifests itself in the physical layer media (wire-pair, wireless, fiber), terminals (telephones, PDAs, workstations), access techniques (time, code, and frequency-division multiplexing), transport assumptions (fixed and variable bit rates), protocols (circuit- and packet-switched), terminal operating systems, and applications (data, audio, video). Past efforts in network design and standardization have embraced heterogeneity in certain forms, particularly in the physical media, but have sought homogeneity at some level (typically transport protocol and establishment protocol). Unfortunately there are several such network designs, including the public telephone network (PTN), the Internet, CATV distribution systems (which are evolving into networks), local area networks, ATM and Frame Relay networks, and a proliferation of wireless networks (DECT, GSM, IS-54 and IS-95, JDC, etc.). Most of these networks are likely to persist for a long time, and there is no credible path toward a single relatively homogeneous network (such as the simpler situation in 1950 when all we had was the PTN). In fact, there is widespread consensus that it is desirable to encourage rapid technological and commercial innovation, a process that rapidly spawns new networking technologies. These different networks can coexist on the same media (such as Internet access on CATV), and can even ride on top of one another (such as IP on ATM or the PTN), all the while remaining logically separate.

Heterogeneity is not benign. The vision in which users seamlessly share multimedia applications over a heterogeneous networking universe will be difficult to realize. Still, users of the Internet should be able to telephone users on the PTN, and users with CATV as their primary network access should be able to videoconference with users on Frame Relay networks, PTN, and the Internet. In short, all general-purpose networks should provide unlimited connectivity to all users, regardless of what networks they access. Of course, applications will be subject to the limitations of the underlying networks, such as bandwidths, error rates, and the like, but interconnection of networks should not, by itself, introduce substantial additional limitations.

Embracing heterogeneity is one of the prime goals of network architecture design. A conceptually simple mechanism for internetworking is to allow each network to use its local protocols, signal compression, and encryption techniques, and to perform conversions at gateways interconnecting the networks. This approach solves the interoperability problems, but it has some limitations:

- ◆ Privacy by end-to-end encryption is precluded wherever encryption will interfere with the conversion function (as in a video or audio compression

transcoding operation). This forces decryption before conversion, which is a serious breach of privacy. In fact, for a link entirely internal to the network, there will not even be a way for the user to confirm that encryption was performed at all.

- ◆ Conversion adds processing delay. For example, digital cellular base stations add on the order of 80 ms for conversion from one speech code to another. Delay is harder to mitigate than other impairments introduced in the network.
- ◆ For continuous-media services, subjective impairments will accumulate. It is also problematic to characterize the accumulation of such subjective impairments across heterogeneous coding techniques, making it difficult or impossible to predict in advance the end-to-end subjective impairments.
- ◆ Conversions require detailed knowledge of the applications to be embedded within the networks. This is a serious breach of modularity and will result in a substantial increase in complexity. It is also a major barrier to the introduction of new services, since they will often require uneconomic global upgrades of equipment.

An alternative is to make networks transparent and force interoperability at the network edge. This approach has been successfully applied to the Internet. It stimulates the rapid introduction and deployment of new services, since they need to be embedded only in those terminals (or access points) desiring the new service. However, end-to-end transparency introduces its own set of difficulties, particularly for nonprogrammable implementations. Joint source/channel coding, which is important for achieving high-traffic capacity on wireless access channels, is difficult to achieve unless the source and channel coding are designed in close coordination. Transparency is also more challenging for multicast networks, for which compatibility with multiple destination terminals must be achieved.

### 13.3.5 Scalability and Configurability

Scalability and configurability of audio, graphics, and video coding algorithms are essential to maintaining interoperability and efficient use of resources. Such scalability also offers a wealth of novel techniques for congestion control by adjusting data rates to the available bandwidth. A key technique in flexible video compression, for example, is multiresolution or scalable coding. Typically, the video source is successively approximated by multiple layers of

coded bit streams (such as the flows mentioned earlier). This allows different terminals to access the same source representation at different bit rates and resolutions. In a multicast context, different representations can be spawned within the network in a generic fashion, without knowledge of the video coding algorithms or compression syntax.

While there has been significant progress in the standardization of video compression (such as H.261 and MPEG), these standards lack several elements that appear essential for the future requirements in multimedia networks. They offer either no scalability to bandwidth (like H.261 and MPEG1) or limited scalability (like MPEG2). Their ability to support layered coding is either missing or very limited, and they therefore do not support multicast transport. Their scalability to network QoS is also missing, generally requiring a very high reliability that will be costly on wireless access channels. (Past compression activity has focused on minimum bit rate, which is not the best criterion on wireless channels, where reliability is also a significant factor.)

Scalability to transport rate, receiver processing, and display capabilities is needed; so too is configuration to the QoS parameters of the transport environment. A given transmit terminal will encounter in a heterogeneous network environment a wide variety of rate and QoS characteristics, differing radically depending on whether there is a wireless access link or not, for example. For a given combination of loss, corruption, and delay parameters, the source coder must configure to achieve the highest subjective quality. Such configurability is not a feature of existing source-coding standards, for they are typically designed with a fixed QoS in mind, such as high reliability (like MPEG) or robustness to high error rates (like digital cellular VCELP voice-coding algorithms).

Related to this issue is joint source/channel coding. The classical Shannon "separation theorem of source and channel coding" is often not applicable in heterogeneous networks, where we cannot assume that the channel is fixed, time invariant, and known. As has long been recognized, the separation theorem and other results of information theory do not adequately address the implications of delay in interactive services. In practice, the highest traffic capacity and subjective quality will be achieved only through the coordination of source coding and channel coding. Such coordination will yield the largest benefit in wireless access links. In a heterogeneous network there is need to coordinate the configurability of both source and channel coding with the requirements of the other. This configurability requires true negotiation in the connection establishment, making call-admission control more complex.

Another way in which scalable video can be incorporated into an existing network with minimal change in the networking elements (e.g., switches,

routers, etc.) is to apply unequal error-protection (UEP) techniques to protect various layers of the scalable video to varying degrees.

CATV, ATM, IP, digital cellular, packet radio, and other networks implement bearer services with different characteristics. It will be necessary to explore the interoperability of these services and its architectural implications. An example that we discussed in section 6.7.3 is the transmission of TCP connections across an ATM network. The question in that situation is when to open and close ATM virtual circuits.

### 13.3.6 Extensibility and Complexity Management

The network must be designed so that it can evolve and adapt over a wide range of parameters as the internetwork grows in size, speed, complexity, and technology. The ability to gracefully accommodate increasing levels of usage and applications with increasing bandwidth requirements is a central challenge of multimedia networking. It is not clear that existing solutions can cost-effectively scale to millions of users at entertainment qualities. At the same time, the multimedia network architectures must accommodate new technologies. One challenge is to define architectural concepts that are inherently scalable and do not limit future possibilities.

The extensibility of a heterogeneous multimedia network bears some resemblance to problems encountered in large software systems, where complexity management has been found to be critical. A fairly small but effective set of management principles has evolved. Two key concepts are modularity (partitioning of functionality into independent and configurable modules) and abstraction (the hiding of irrelevant implementation details, with explicit visibility of only essential characteristics). These principles serve to separate the system into modules that can evolve independently by avoiding unnecessary dependence among them. Independent evolution is essential to extensibility because global upgrades are not economically feasible.

When applying the principles of complexity management to multimedia networks, legitimate and necessary dependencies must be taken into account, while making them as benign as possible through appropriate configuration flexibility. For example, the joint source/channel coding mentioned earlier creates a necessary but undesirable dependency between the design of the source coder and the channel coder. We must be sure that this dependency is not hard-wired into the system (as is the case with many existing standards, such as digital cellular voice), but is introduced in a generic and configurable

fashion that can accommodate future unanticipated applications. This may be achieved through the abstraction of the channel to appropriate QoS models, explicitly hiding details such as wired or wireless transport, and the like. In addition, there must be configurability through a true negotiation of QoS parameters as mentioned earlier.

A key to extensibility is sufficient flexibility in the establishment phase to allow future evolution of the network without massive changes or upgrades. One approach is to use software agents as part of the establishment protocols. In this approach, the traditional static message structures are replaced by dynamic executable messages that embed procedures and associated dynamic data structures. In essence, one module configures another (or more generally conducts a negotiation) by sending an executable message. This increases flexibility, because static message sets limit the semantics to the description of an operating point, whereas agents can specify an individual control interface for each network component. This approach can also dramatically reduce the delay caused by the establishment negotiations. The problem with conventional approaches is that an establishment negotiation may require numerous back-and-forth messages, incurring considerable latency. In agent-based establishment, the different network elements or subnetworks and the requirements of the network can be represented by distinct agents collected in a common processor for negotiation, eliminating communication latency while maintaining desirable logical modularity. Object-oriented Tcl (Tool Command Language) and Java can facilitate the description of agents.

Extensibility to new services and applications is also a critical feature of future multimedia networks. Extensibility will be facilitated by the ability to prototype and deploy new services with minimal effort. Rapidly deployable applications can be based on a service description language, a platform at each terminal incorporating the networked operating system, an interpreter of the service language, and a collection of resources or primitives for the realization of services. These resources include hardware and processing capability, as well as stored software definitions of useful service elements, such as audio or video compression algorithms. New services can be deployed by transferring a service description agent during call setup. Such descriptions are likely to be large.

This agent approach to service deployment avoids two traditional obstacles. The first is the critical size problem, where a sufficient number of specialized terminals must be deployed before an economically sustainable community of service participants exists. (The critical size problem was described in section 1.2.) In the agent approach, there must exist only a critical mass of

platforms, not specialized service terminals. The second obstacle is standardization. While we require standardization of the platform, service description language, and method of transferring service descriptions in call setup, we avoid standardization of specific new services before they are deployed. An additional advantage of the agent approach is that services can be dynamically reconfigured during a call.

### 13.3.7 Security

The network is a shared resource, providing the advantages of access to a wide community of users and services. This brings with it the disadvantage of potential breaches of security, lack of privacy, and exposure to fraud. Security has many components. These include confidentiality and integrity (inability of unauthorized parties to read or modify information), as well as authentication and nonrepudiation (providing the equivalent of an electronic signature). It may be desirable in some cases even to mask the fact that communication is taking place between a subset of users (masking of traffic patterns). Multimedia networking poses special security questions because of the interaction among compression, encryption, and error propagation.

Encryption hides the basic syntactical and semantic elements of a bit stream and thus obstructs many important processes, such as protocol conversion, standard conversion, and joint source/channel coding. The proper placement of encryption within the network architecture is an important question. Deploying encryption at lower layers such as the network layer or data link layer allows encryption of routing overhead from higher layers, thus masking traffic patterns better. However, at internetwork gateways, such as between OSI and TCP/IP, the user data has to be decrypted and reencrypted, making it vulnerable to eavesdropping. Deploying encryption at higher layers, such as at the application layer, has the advantage that user data can be encrypted end to end. But then headers appended by lower layers, being unencrypted, give clues to the traffic patterns. Further, the number of entities that need to be separately encrypted is much larger, as each of the user processes associated with the application now needs to be encrypted. These trade-offs are strongly influenced by the way the applications are structured.

Audio and video can tolerate residual errors at playback or display, and thus error propagation in encryption is important. Further, if multiscale representations of information are used as suggested above, it will be important to structure encryption so that error propagation preserves the hierarchical structure of the information.

In more complicated multiuser, multicast applications (for example, video-conferencing) the very definition of privacy becomes problematic. For instance, managers in a conference with labor might wish to briefly have a private conversation that cannot be heard by the labor representatives (or vice versa). This illustrates the importance of considering application requirements, including privacy, in the definition of network service primitives.

---

# Bibliography

- [A93] The ATM Forum (1993). *ATM User-Network Interface Specification: Version 3.0*. PTR Prentice Hall, Englewood Cliffs, NJ.
- [A94] A. S. Acampora (1994). *An Introduction to Broadband Networks: LANs, MANs, ATM, B-ISDN, and Optical Networks for Integrated Multimedia Telecommunications*. Plenum, New York.
- [A95a] The ATM Forum (1995). *BISDN Inter Carrier Interface (B-ICI) Specification: Version 2.0 (Integrated)*.
- [A95b] The ATM Forum (1995). *LAN Emulation over ATM: Version 1.0*.
- [A96a] The ATM Forum (1996). *ATM User-Network Interface (UNI) Signalling Specification: Version 4.0*.
- [A96b] The ATM Forum (1996). *Integrated Local Management Interface (ILMI) Specification: Version 4.0*.
- [A96c] The ATM Forum (1996). *Private Network-Network Interface Specification: Version 1.0 (PNNI 1.0)*.
- [A96d] The ATM Forum (1996). *Traffic Management Specification: Version 4.0*.
- [A97a] The ATM Forum (1997). *Frame-Based User-to-Network Interface Specification v2.0*.
- [A97b] The ATM Forum (1997). *Multi-Protocol over ATM: Version 1.0*.
- [A99] The ATM Forum (1999). *ATM Forum Addressing: Reference Guide*.
- [AA73] R. Artle and C. Averous (1973). The telephone system as a public good. *Bell Journal of Economics and Management Science* 4(1):89–100.
- [Ab96] N. Abramson (1996). Wide-band random-access for the last mile. *IEEE Personal Communications Magazine* 3(6):29–33.

- [ABB96] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor (1996). Adaptive mobile multimedia networks. *IEEE Personal Communications Magazine* 3(2):34–51.
- [AL95] A. Alles (1995). ATM Internetworking. Cisco Systems, <http://www.cisco.com/>, May 1995.
- [ALFV98] J. D. Angelopoulos, N. I. Lepidas, E. K. Fragoulopoulos, and I. S. Venieris (1998). TDMA multiplexing of ATM cells in a residential access SuperPON. *IEEE J. Selected Areas in Communications* 16(7):1123–1133.
- [ALM98] G. Anastasi, L. Lenzini, and E. Mingozzi (1998). MAC protocols for wideband wireless local access: evolution toward wireless ATM. *IEEE Personal Communications Magazine* 5(5): 53–64.
- [AMS82] D. Anick, D. Mitra, and M. M. Sondhi (1982). Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal* 61(8):1871–1894, October 1982.
- [AS96] F. Abrishamkar and Z. Siveski (1996). PCS global mobile satellites. *IEEE Communications Magazine* 34(9):132–136.
- [ATM1] The ATM Forum (1995). ATM Forum 94-0471R7: P-NNI draft specification, March 1995.
- [ATM2] The ATM Forum (1995). LAN emulation over ATM specification—Version 1, February 1995.
- [B90] J. A. C. Bingham (1990). Multicarrier modulation for data transmission: An idea whose time has come. *IEEE Communications Magazine* 28(5):5–14.
- [B95] K. Balaji (1995). *Broadband Communications: A Professional's Guide to ATM, Frame Relay, SMDS, SONET, and B-ISDN*. McGraw-Hill, New York.
- [B99] U. Black (1999). *ATM: Foundation for Broadband Networks*. 2d ed. Vol. 1. Prentice Hall, Upper Saddle River, NJ.
- [BC89] R. Ballart and Y. C. Ching (1989). SONET: Now it's the standard optical network. *IEEE Communications Magazine* 27(3):8–15.
- [BCB96] K. Bala, F. R. K. Chung, and C. A. Brackett (1996). Optical wavelength routing, translation, and packet/cell switched networks. *Journal of Lightwave Technology* 14(3):336–343.
- [BD94] D. D. Botvich and N. G. Duffield (1994). Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. Preprint.
- [BG92] D. Bertsekas and R. Gallager (1992). *Data Networks*. Prentice Hall, Englewood Cliffs, NJ.
- [BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima (1993). Near Shannon limit error-correcting coding: Turbo codes. *Proceedings of IEEE International Conference on Communications '93*, pp. 1064–1070.
- [BP79] C. A. Belfiore, Jr., and J. H. Park (1979). Decision-feedback equalization. *IEEE Proceedings* 67(8):1143–1156.

- [BPS98] E. Biglieri, J. Proakis, and S. Shamai (1998). Fading channels: Information theoretic and communications aspects. *IEEE Transactions on Information Theory* 44(6): 2619–2692.
- [BPSK97] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz (1997). A comparison of mechanisms for improving tcp performance over wireless links. *IEEE/ACM Transactions on Networking* (December), 756–769.
- [BR60] R. R. Bahadur and R. R. Rao (1960). On deviations of the sample mean. *Ann. Math. Statist.* 31(1960):1015–1027.
- [Bu90] J. A. Bucklew (1990). *Large Deviation Techniques in Decision, Simulation, and Estimation*. John Wiley & Sons, New York.
- [C85] L. J. Cimini, Jr. (1985). Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Transactions on Communications* 33(7):665–675.
- [C88] D. Comer (1988). *Internetworking with TCP/IP: Principles, Protocols, and Architecture*. Prentice Hall, Englewood Cliffs, NJ.
- [C89] R. L. Carroll (1989). Optical architecture and interface lightguide unit for fiber-to-the-home feature of the AT&T SLC series 5 carrier system. *Journal of Lightwave Technology* 7(11):1727–1732.
- [C90] J. A. Chiddix (1990). Fiber backbone trunking in cable television networks: An evolutionary adoption of new technology. *IEEE LCS Magazine* 1(1):32–37.
- [C91] R. L. Cruz (1991). A calculus for network delay, I. Network elements in isolation, II. Network analysis. *IEEE Transactions on Information Theory* 37(1):114–131, 132–141.
- [C95] D. Comer (1995). *Internetworking with TCP/IP*. Prentice Hall, Englewood Cliffs, NJ.
- [CGK92] I. Chlamtac, A. Ganz, and G. Karmi (1992). Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Transactions on Communications* 40(7):1171–1182.
- [CJ89] D. Chiu and R. Jain (1989). Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems* 17(1):1–14.
- [CK74] V. G. Cerf and R. E. Kahn (1974). A protocol for packet network intercommunication. *IEEE Transactions on Communications* 22(5):647–648.
- [CKD98] I. Chrisment, D. Kaplan, and C. Diot (1998). An ALF communication architecture: design and automated implementation. *IEEE J. Selected Areas in Communications* 16(3):332–344.
- [Cl88] D. Clark (1988). The design philosophy of the DARPA internet protocols. *Proceedings of the SIGCOMM '88 Symposium*, 106–114.
- [ClT90] D. Clark and D. Tennerhouse (1990). Architectural considerations for a new generation of protocols. *Proceedings of the SIGCOMM '90 Symposium*, 106–114.
- [CM91] *IEEE Communications Magazine* (1991). The 21st Century Subscriber Loop, vol. 29, no. 3.
- [CM92] *IEEE Communications Magazine* (1992). Intelligent Networks, vol. 31, no. 2.
- [CM94] *IEEE Communications Magazine* (1994). Video on Demand, vol. 32, no. 5.

- [CM95] *IEEE Communications Magazine* (1995). Access to Broadband Services, vol. 33, no. 8.
- [CM98a] *IEEE Communications Magazine* (1998). Deployment of WDM Fiber-based Optical Networks, vol. 36, no. 2.
- [CM98b] *IEEE Communications Magazine* (1998). WDM Fiber Optic Communications, vol. 36, no. 12.
- [CMM94] A. Claessen, L. Monteban, and H. Moelard (1994). The AT&T GIS WaveLAN air interface and protocol stack. *Proceedings of Wireless Networks: Catching the Mobile Future*, 1442–1446.
- [CN95] CommerceNet and Nielsen Media Research (1995). *The CommerceNet/Nielsen Internet Demographics Survey: Executive Summary*. Available at: <http://www.commerce.net>.
- [COUCR97] G. Cherubini, S. Olcer, G. Ungerboeck, J. Creigh, and S. K. Rao (1997). 100BASE-T2: a new standard for 100 Mb/s Ethernet transmission over voice-grade cables. *IEEE Communications Magazine* 35(11):115–122.
- [CT91] T. M. Cover and J. A. Thomas (1991). *Elements of Information Theory*. John Wiley & Sons, New York.
- [CW89] D. R. Cheriton and C. L. Williamson (1989). VMTP as the transport layer for high performance distributed systems. *IEEE Communications Magazine* 27(6):37–44.
- [CW94] W. Y. Chen and W. L. Waring (1994). Applicability of ADSL to support video dial tone in the copper loop. *IEEE Communications Magazine* 32(5):102–109.
- [CW96] C. Courcoubetis and R. Weber (1996). Buffer overflow asymptotics for a buffer handling many traffic sources. *Journal of Applied Probability* 33(3):886–903.
- [CWSA95] J. Crowcroft, Z. Wang, A. Smith, and J. Adams (1995). A rough comparison of the IETF and ATM service models. *IEEE Network* 9(6):12–16.
- [CWW96] C. Courcoubetis, J. Walrand, and R. Weber (1996). Pricing models for multiclass networks. In preparation.
- [CZA93] R. Chipalkatti, Z. Zhang, and A. S. Acampora (1993). Protocols for optical star-coupler network using WDM: performance and complexity studies. *IEEE J. Selected Areas in Communications* 11(4):579–589.
- [D91] R. Durrett (1991). *Probability: Theory and Examples*. Wadsworth and Brooks/Cole, Pacific Grove, CA.
- [D94] R. C. Dixon (1994). *Spread Spectrum Systems with Commercial Applications*. 3rd ed. John Wiley & Sons, New York.
- [DDR98] B. Davie, P. Doolan, and Y. Rekhter (1998). *Switching in IP Networks*. Morgan Kaufmann, San Francisco.
- [DE96] S. Dixit and S. Elby (1996). Frame Relay and ATM internetworking. *IEEE Communications Magazine* 34(6):64–82.
- [DGM94] D. A. Dunn, W. D. Grover and M. H. MacGregor (1994). Comparison of  $k$ -shortest paths and maximum flow routing for network facility restoration. *IEEE J. Selected Areas in Communications* 12(1):88–99.

- [DJB95] C. Douillard, M. Jezequel, and C. Berrou (1995). Iterative correction of intersymbol interference: Turbo equalization. *Europ. Trans. Telecommun.*, pages 507–511.
- [DV93] G. de Veciana (1993). *Design Issues in ATM Networks: Traffic Shaping and Congestion Control*. Ph.D. thesis, Dept. of EECS, University of California, Berkeley.
- [DZ93] A. Dembo and O. Zeitouni (1993). *Large Deviations Techniques and Applications*. Jones and Bartlett, Boston.
- [E99] R. Edell (1999). *INDEX: Internet Demand Experiment*. Ph.D. thesis, Dept. of EECS, University of California, Berkeley.
- [EMV95] R. Edell, N. McKeown, and P. P. Varaiya (1995). Billing users and pricing for TCP. *IEEE J. Selected Areas in Communications* 13(7):1162–1175.
- [F72] G. D. Forney, Jr. (1972). Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory* 18(3):363–378.
- [F95] D. Frankel (1995). ISDN reaches the market. *IEEE Spectrum* 32(6):20–25.
- [F98] H. Frazier (1998). The 802.3z gigabit Ethernet standard. *IEEE Network* 12(3):6–7.
- [FJ91] S. Floyd and V. Jacobson (1991). Traffic phase effects in packet-switched gateways. *Computer Communication Review* 21(2):26–42.
- [FJ93] S. Floyd and V. Jacobson (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking* 1(4):397–413.
- [FMM91] K. W. Fendick, D. Mitra, I. Mitrani, M. A. Rodriguez, J. B. Seery, and A. Weiss (1991). An approach to high-performance, high-speed data networks. *IEEE Communications Magazine*, October 1991, 74–82.
- [G93] P. E. Green, Jr. (1993). *Fiber Optic Networks*. Prentice Hall, Englewood Cliffs, NJ.
- [G96] P. E. Green, Jr. (1996). Optical networking update. *IEEE J. Selected Areas in Communications* 14(5):764–779.
- [GC97] A. J. Goldsmith and S.-G Chua (1997). Variable-rate variable-power M-QAM for fading channels. *IEEE Transactions on Communications* 46(5):1218–1230.
- [GCMW99] R. Gupta, M. Chen, S. McCanne, and J. Walrand (1999). WebTP: A receiver-driven web transport protocol. *Proc. IEEE INFOCOM*.
- [GG92] A. Gersho and R. M. Gray (1992). *Vector Quantization and Signal Compression*. Kluwer, Norwell, MA.
- [GH91] R. J. Gibbens and P. J. Hunt (1991). Effective bandwidths for the multi-type UAS channel. *Queueing Systems* 9(1):17–28.
- [GJF98] R. Goyal, R. Jain, S. Fahmy, et al. (1998). Design issues for providing minimum rate guarantees to the ATM unspecified bit rate service. *Proceedings 1998 IEEE ATM Workshop*, 169–175.
- [GK98] R. J. Gibbens and F. P. Kelly (1998). Resource pricing and the evolution of congestion control. Statistical Laboratory Research Report 1998-10. Cambridge University, U.K.

- [GS99] S. J. Golestani and K. K. Sabnani (1999). Fundamental observations on multicast congestion control in the Internet. *Proc. IEEE INFOCOM*.
- [H29] H. Hotelling (1929). Stability in competition. *Economic Journal* 39:41–57.
- [H88] F. Halsall (1988). *Data Communications, Computer Networks and OSI*. Addison-Wesley, Reading, MA.
- [H90] J. Hui (1990). *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer, Norwell, MA.
- [H95] I. Hsu (1995). *Admission Control and Resource Management for Multi-Service ATM Networks*. Ph.D. thesis, Dept. of EECS, University of California, Berkeley.
- [H97] G. T. Hawley (1997). System considerations for the use of xDSL technology for data access. *IEEE Communications Magazine* 35(3):56–60.
- [HA87] J. Y. Hui and E. Arthurs (1987). A broadband packet switch for integrated transport. *IEEE J. Selected Areas in Communications* 5(8):1264–1273.
- [HKR96] E. Hall, J. Kravitz, R. Ramaswami, M. Halvoren, S. Tenbrink, and R. Thomsen (1996). *IEEE J. on Selected Areas in Communications* 14(5):814–823.
- [Hu88] J. Y. Hui (1988). Resource allocation for broadband networks. *IEEE J. Selected Areas in Communications* 6(9):1598–1608.
- [HW94] I. Hsu and J. Walrand (1994). Admission control for ATM networks. *Proceedings of the IMA Workshop on Stochastic Networks*, March 1994. IMA volumes in Mathematics and Its Applications, vol. 71, 411–427. Springer-Verlag (1995).
- [HW99] C. Heegard and S. B. Wicker (1999). *Turbo Coding*. Kluwer Academic Publishers, Norwell, MA.
- [I120] ITU-T Recommendation I.120, Integrated services digital network (ISDN). 1992 (rev).
- [IN95] *IEEE Network* (1995). Digital Interactive Broadband Video Dial Tone Networks, vol. 9, no. 5.
- [J83] Y.-C. Jenq (1983). Performance analysis of a packet switch based on single-buffered banyan network. *IEEE J. Selected Areas in Communications* 1(6):1014–1021.
- [J88] V. Jacobson (1988). Congestion avoidance and control. *Computer Communications Review* 18(4): 314–329.
- [J96] R. Jain (1996). Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems* 28(13):1723–1728.
- [JBM96] J. P. Jue, M. S. Borella, and B. Mukherjee (1996). Performance analysis of the Rainbow WDM optical network prototype. *IEEE J. Selected Areas in Communications* 14(5):945–951.
- [JM96] D. Johnson and D. Maltz (1996). Protocols for adaptive wireless and mobile networking. *IEEE Personal Communications Magazine* 3(1):34–42.
- [JSAC95] *IEEE Journal on Selected Areas in Communications* (1995). Copper Wire Access Technologies for High Performance Networks, vol. 13, no. 9.

- [JSAC96] *IEEE Journal on Selected Areas in Communications* (1996). Optical Networks, vol. 14, no. 5.
- [JSAC98] *IEEE Journal on Selected Areas in Communications* (1998). High-Capacity Optical Transport Networks, vol. 16, no. 7.
- [JV91] S. Jordan and P. P. Varaiya (1991). Throughput in multiple service, multiple resource communication networks. *IEEE Transactions on Communications* 39(8):1216–1222.
- [JV94] S. Jordan and P. P. Varaiya (1994). Control of multiple service, multiple resource communication networks. *IEEE Transactions on Communications* 42(11):2979–2988.
- [K75] L. Kleinrock (1975). *Queueing Systems*. John Wiley & Sons, New York.
- [K79] F. P. Kelly (1979). *Reversibility and Stochastic Networks*. John Wiley & Sons, New York.
- [K91] F. P. Kelly (1991). Effective bandwidths at multi-class queues. *Queueing Systems*, no. 9:5–16.
- [K94] F. P. Kelly (1994). Dynamic routing in stochastic networks. IMA volumes in *Mathematics and Its Applications*, ed. F. P. Kelly and R. J. Williams, vol. 71, 169–186. Springer-Verlag (1995).
- [KB97] J. M. Kahn and J. R. Barry (1997). Wireless infrared communications. *IEEE Proceedings* 85(2):265–298.
- [KDD96] I. P. Kaminow, C. R. Doerr, C. Dragone, et al. (1996). A wideband all-optical WDM network. *IEEE J. Selected Areas in Communications* 14(5):780–799.
- [KHM87] M. J. Karol, M. G. Hluchyj, and S. P. Morgan (1987). Input versus output queueing on a space-division packet switch. *IEEE Transactions on Communications* 25(12):1347–1356.
- [KL90] H. S. Kim and A. Leon-Garcia (1990). Performance of buffered banyan networks under nonuniform traffic patterns. *IEEE Transactions on Communications* 38(5):648–658.
- [Kle94] L. Kleinrock (1994). *Realizing the Future: The Internet and Beyond*. National Academy Press, Washington, D.C.
- [KN96] I. Katzela and M. Naghshineh (1996). Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. *IEEE Personal Communications Magazine* 3(3):10–22.
- [KS83] C. P. Kruskal and M. Snir (1983). The performance of multistage interconnection networks for multiprocessors. *IEEE Transactions on Computers* C-32(12):1091–1098.
- [KWC93] G. Kesidis, J. Walrand, and C. S. Chang (1993). Effective bandwidths for multiclass Markov fluids and other ATM sources. *IEEE/ACM Transactions on Networking* (August), 424–428.
- [L90] T. T. Lee (1990). Modular architecture for very large packet switches. *IEEE Transactions on Communications* 38(7):1097–1106.
- [LC + 97] B. M. Leiner, V. G. Cerf, et al. (1997). The past and future history of the Internet. *Communications of the ACM* 40(2):102–108.
- [LL91] S. C. Liew and K. W. Lu (1991). Comparison of buffering strategies for asymmetric packet switch modules. *IEEE J. Selected Areas in Communications* 9(3):428–438.

- [LLKS95] H. C. Lucas, Jr., H. Levecq, R. Kraut, and L. Streeter (1995). France's grass-roots data net. *IEEE Spectrum* 32(11):71–77.
- [LN97] S. Lin and N. McKeown (1997). A simulation study of IP switching. *Computer Communication Review* 27(4):15–24.
- [LNT87] B. Leiner, N. Nielson, and F. Tobagi (1987). Issues in packet radio design. *IEEE Proceedings* 75(1):6–20.
- [LV95] S. Low and P. Varaiya (1995). Burst reducing servers in ATM networks. *Queueing Systems* 20:61–84.
- [M79] V. H. McDonald (1979). The cellular concept. *Bell Systems Technical Journal* (January), 15–49.
- [M92a] B. Mukherjee (1992). WDM-based local lightwave networks. Part I: Single-hop systems. *IEEE Networks* (May), 12–27.
- [M92b] B. Mukherjee (1992). WDM-based local lightwave networks. Part I: Multi-hop systems. *IEEE Networks* (July), 20–32.
- [M95] D. J. G. Mestdagh (1995). *Fundamentals of multiaccess optical fiber networks*. Artech House, Boston.
- [M98] M. W. Maeda (1998). Management and control of transparent optical networks. *IEEE J. Selected Areas in Communications* 16(7):1008–1023.
- [M99] E. Modiano (1999). WDM-based packet networks. *IEEE Communications Magazine* (March), 130–135.
- [MADD98] J. Manchester, J. Anderson, B. Doshi, and S. Dravida (1998). IP over SONET. *IEEE Communications Magazine* 36(5):136–143.
- [MV95] J. K. MacKie-Mason and H. R. Varian (1995). Pricing congestible network resources. *IEEE J. Selected Areas in Communications* 13(7):1141–1149.
- [MVW93] N. McKeon, P. Varaiya, and J. Walrand (1993). Scheduling cells in an input-queued switch. *Electronics Letters* 29(25):2174–2175, December.
- [MW96] M. Molle and G. Watson (1996). 100Base-T/IEEE 802.12/Packet switching. *IEEE Communications Magazine* 34(8):64–73.
- [MW98] J. Mo and J. Walrand (1998). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*. Submitted.
- [OMKM89] Y. Oie, M. Murata, K. Kubota, and H. Miyahara (1989). Effect of speedup in nonblocking packet switch. *Proceedings of IEEE International Conference on Communications '89*, pp. 410–414.
- [OP98] T. Ojanpera and R. Prasad (1998). An overview of air interface multiple access for IMT-2000/UMTS. *IEEE Communications Magazine* 36(9):82–95.
- [P81] J. H. Patel (1981). Performance of processor-memory interconnections for multiprocessors. *IEEE Transactions on Computers* C-30(10):771–780.
- [P92] D. Parsons (1992). *The Mobile Radio Propagation Channel*. John Wiley & Sons, New York.

- [P93] M. Padovano (1993). *Networking Applications on UNIX System V Release 4*. Prentice Hall, Englewood Cliffs, NJ.
- [P94] C. Partridge (1994). *Gigabit Networking*. Addison-Wesley, Reading, MA.
- [P95] G. Pettersson (1995). ISDN: From custom to commodity service. *IEEE Spectrum* 32(6):26–31.
- [P96] C. Perkins (1996). IP mobility support. IETT Internet Draft, February 9, 1996.
- [PB95] W. Pugh and G. Boyer (1995). Broadband access: Comparing alternatives. *IEEE Communications Magazine* 33(8):34–47.
- [PC96] *IEEE Personal Communications Magazine* (1996). Special issue on Wireless ATM, vol. 3, no. 4.
- [PC98] *IEEE Personal Communications Magazine* (1998). Special issue on Smart Antennas, vol 5, no. 1.
- [PL95] K. Pahlavan and A. H. Levesque (1995). *Wireless Information Networks*. John Wiley & Sons, New York.
- [Pr95] J. G. Proakis (1995). *Digital Communications*. 3rd ed. McGraw-Hill, New York.
- [R91] T. R. Rowbotham (1991). Local loop development in the U.K. *IEEE Communications Magazine* 29(3):50–59.
- [R93] T. G. Robertazzi, ed. (1993). *Performance Evaluation of High Speed Switching Fabrics and Networks*. IEEE Press, Piscataway, NJ.
- [R96] T. S. Rappaport (1996). *Wireless Communications: Principles and Practice*. PTR Prentice Hall, Upper Saddle River, NJ.
- [RM98] B. Ramamurty and B. Mukherjee (1998). Wavelength conversion in WDM networking. *IEEE J. Selected Areas in Communications* 16(7):1061–1073.
- [RS96] S. Ramanathan and M. Steenstrup (1996). A survey of routing techniques for mobile communication networks. *J. Special Topics in Mobile Networks and Applications (MONET)* 1(2):89–104.
- [RS97] R. Ramaswami and K. N. Sivarajan (1997). *Optical Networks*. Morgan-Kaufmann, San Francisco.
- [RW98] P. Robertson and T. Worz (1998). Bandwidth-efficient turbo trellis-coded modulation using punctured component codes. *IEEE J. Selected Areas in Communications* 16(2):206–218.
- [S92] W. Stallings (1992). *ISDN and Broadband ISDN*. Macmillan, New York.
- [S96] K. Sayood (1996). *Introduction to Data Compression*. Morgan Kaufmann, San Francisco.
- [SHL99] J. M. Senior, M. R. Handley, and M. S. Leeson (1999). Developments in wavelength division multiple access networking. *IEEE Communications Magazine* 36(12):28–36.
- [So96] S. J. B. Yoo (1996). Wavelength conversion technologies for WDM network applications. *Journal of Lightwave Technology* 14(6):955–966.
- [St96] G. Stuber (1996). *Principles of Mobile Communication*. Kluwer, Boston.

- [SV99] C. Shapiro and H. R. Varian (1999). *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston.
- [SW95] A. Schwartz and A. Weiss (1995). *Large Deviations for Performance Analysis*. Chapman and Hall, New York.
- [T88] A. S. Tanenbaum (1988). *Computer Networks*. 2d ed. Prentice Hall, Englewood Cliffs, NJ.
- [TG87] P. Temin and L. Galambos (1987). *The Fall of the Bell System: A Study in Prices and Politics*. Cambridge University Press, New York.
- [TO98] N. Takachio and S. Ohteru (1998). Scale of WDM transport network using different types of fibers. *IEEE J. Selected Areas in Communications* 16(7):1320–1326.
- [U82] G. Ungerboeck (1982). Channel coding with multi-level/phase signals. *IEEE Transactions on Information Theory* 28(1):55–67.
- [V93] H. R. Varian (1993). *Intermediate Microeconomics: A Modern Approach*. W. W. Norton, New York.
- [V95] A. J. Viterbi (1995). *CDMA Principles of Spread Spectrum Communications*. Addison-Wesley, New York.
- [V98] S. Verdú (1998). *Multiuser Detection*. Cambridge University Press, Cambridge, U.K.
- [VVM93] W. Verbist, G. Van der Plas, and D. J. G. Mestdagh (1993). FITL and B-ISDN: A marriage with a future. *IEEE Communications Magazine* 31(6):60–66.
- [W86] A. Weiss (1986). A new technique for analyzing large traffic systems. *Adv. Appl. Prob.*, 506–532.
- [W88] J. Walrand (1988). *An Introduction to Queuing Networks*. Prentice Hall, Englewood Cliffs, NJ.
- [W95] A. Weiss (1995). An introduction to large deviations for communication networks. *IEEE J. Selected Areas in Communications* 13(6):938–952.
- [W98] J. Walrand (1998). *Communication Networks: A First Course*. 2d ed. WCB/McGraw-Hill.
- [Wi98] J. H. Winters (1998). Smart antennas for wireless systems. *IEEE Personal Communications Magazine* 5(1):23–27.
- [WL89] S. S. Wagner and H. L. Lemberg (1989). Technology and system issues for the WDM-based fiber loop architecture. *Journal of Lightwave Technology* 7(11):1759–1768.
- [WL92] T.-H. Wu and R. C. Lau (1992). A class of self-healing ring architectures for SONET network applications. *IEEE Transactions on Communications* 40(11):1746–1756.
- [Wu95] T.-H. Wu (1995). Emerging technologies for fiber network survivability. *IEEE Communications Magazine* 33(2):60–74.
- [X92] XTP Forum (1992). *Xpress Transfer Protocol Version 4.0*. Technical Report. [Ptp://dancer.ca.sandia.gov/pub/xtp4.0/xtp4.0-specification-25.ps](http://dancer.ca.sandia.gov/pub/xtp4.0/xtp4.0-specification-25.ps)
- [YHA87] Y.-S. Yeh, M. G. Hluchyj, and A. S. Acampora (1987). The knockout switch: A simple modular architecture for high performance packet switch. *IEEE J. Selected Areas in Communications* 5(8):1274–1283.

- [YLL90] H. Yoon, K. Y. Lee, and M. T. Liu (1990). Performance analysis of multibuffered packet-switching networks in multiprocessor systems. *IEEE Transactions on Computers* 39(3):319–327.
- [ZDESZ93] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala (1993). RSVP: A new resource ReSerVation protocol. *IEEE Network* 7(5):8–18.

---

# Index

## — A —

- AAL (ATM adaptation layer), 277, 282–285
- ABP (Alternating Bit Protocol), 72–75
- ABR (available bit rate)
  - ATM, 260–262
  - GCRA, 395, 397–398
- absorption, fiber attenuation from, 547
- absorption costs, 493
- abstraction, 639
- access control, 517
- access fees, 506–507
- access lines, 57
- ACK flag, 179–180
- ack packets, 165
- acknowledgments
  - ABR, 73–74
  - in error control, 71
  - Go Back N, 75–76
  - random access protocols, 336–337
  - reliable multicast, 175
  - Selective Repeat Protocol, 77
  - TCP, 179, 189
  - TFTP, 183
- active networks, 632
- active queue management, 191
- ad hoc wireless networks, 355–356
- adaptive antennas, 331
- adaptive coding, 632
- adaptive modulation, 327–328
- add/drop multiplexers (ADMs)
  - DWDM, 224
  - optical, 558
  - SONET, 213–214
- Add Party messages, 272
- additive increase and multiplicative decrease algorithm, 390
- Address Registration MIB module, 291
- address resolution for LANE, 297
- Address Resolution Protocol (ARP), 168, 295
- addresses
  - ATM, 269–270
  - circuit-switched networks, 208
  - Ethernet packets, 118
  - IP protocols, 159, 163–167
  - IPv6, 176–177
  - MAC, 118, 123, 126, 163–164, 269, 297
  - multicast IP, 173–174

- addresses (*cont.*)  
  packet switching, 67, 589–590  
  routing table searches, 592
- adjacent channel interference, 321–322
- adjacent IMEs, 291
- administration costs, 24
- admission control, 365–366  
  ATM, 268, 398–403  
  in routing optimization, 378  
  RSVP, 194
- ADMs (add/drop multiplexers)  
  DWDM, 224  
  optical, 558  
  SONET, 213–214
- ADSL (Asymmetric Digital Subscriber Line) service, 10, 235–239
- ADSL Terminal Units (ATUs), 235
- ADUs (application data units), 195
- advertisement messages, 176
- advertising, site rents for, 494
- agent advertisement messages, 176
- agent solicitation messages, 176
- aggregation costs, 496
- aggregation of QoS, 629–630
- AIS (alarm indication signal) field, 289
- ALF (application-level framing), 195
- AlGaAs (aluminum gallium arsenide)  
  lasers, 550
- allocation control, 366
- ALOHA techniques, 334–335
- Alohanet network, 20, 306–307
- Alternating Bit Protocol (ABP), 72–75
- aluminum gallium arsenide (AlGaAs)  
  lasers, 550
- amplifiers and amplification  
  fiber, 549–550  
  lasers, 544  
  optical receivers, 544–545  
  WDM systems, 556–557
- analog signals  
  digitization of, 21–24  
  telephone networks, 6–7
- antennas, 331
- anycast service, 270
- aperiodic probability transition matrices, 437
- APON (ATM over PON) systems, 229
- application data units (ADUs), 195
- application layer in OSI model, 113
- application-level framing (ALF), 195
- applications, 41  
  audio and video streams, 42–43  
  client/server, 43–44  
  global multimedia networks, 626  
  improvements needed in, 92  
  Internet, 185  
  networked games, 43  
  TCP/IP, 181–182  
  voice over packets and  
    videoconferences, 43
- World Wide Web, 42
- applications layer in ODN model, 86–88
- architectures  
  global multimedia networks, 623–624, 627–628  
  layered, 80–86  
  network, 89–91  
  wireless networks, 337–339
- ARDIS network, 351
- ARP (Address Resolution Protocol), 168, 295
- ARPANET network, 13, 156
- ARQ (Automatic Repeat Request) protocol, 326, 368
- ASs (autonomous systems), 172–173
- associative memory, 592
- Asymmetric Digital Subscriber Line (ADSL) service, 10, 235–239
- asynchronous traffic, 134–135

- Asynchronous Transfer Mode (ATM)  
networks, 16, 257–258  
addressing, 269–270  
ATM adaptation layer, 282–285  
ATM Forum recommendations, 395–398  
averaging rate fluctuations, 407–408  
BISDN in, 293–294  
buffering, 413–419  
burstiness, 468–469  
as connection-oriented service, 259–262  
decoupling bandwidth, 480  
effective bandwidth, 476–478  
features, 258–259  
fixed cell size, 262–266  
future, 29  
GCRA for, 419–421  
header structure, 277–278  
other fields in, 280–281  
reserved VCI/VPI in, 281–282  
VCIs and VPIs in, 278–280  
internetworking with, 294–295  
Frame Relay and SMDS over ATM, 301  
IP over ATM, 297–300  
LAN emulation over ATM, 295–296  
multiprotocol over AAL5, 295  
multiprotocol over ATM, 300–301  
leaky bucket, 394, 467–468  
linear bounds, 466–467  
management and control, 285–287, 392–393, 465–466  
Bahadur-Rao theorem, 480–482  
control problems, 393–395  
deterministic approaches, 395–405, 466–469  
deterministic vs. statistical procedures, 422–424  
fault management, 287–290  
large variations of random variables iid, 470–474  
network status monitoring and configuration, 291–292  
queue large deviations, 475–480  
statistical procedures, 405–422  
straight-line large deviations, 474–475  
traffic and congestion control, 290  
user/network signaling, 292  
Markov-modulated fluids for, 405–406  
multiclass case, 419  
multiplexing, 266–267, 408–410, 416–419  
on-line estimation, 410–413  
PNNI routing, 272–276  
pricing for services, 528  
resources and services model, 529–533  
revenue maximization, 533–535  
queuing analysis, 266  
resource allocation, 267–269  
signaling, 270–272  
switches, 2, 266–267  
traffic models, 405  
traffic shaping, 421–422  
wireless, 354–355  
ATM adaptation layer (AAL), 277, 282–285  
ATM Forum  
recommendations, 395–398  
service categories specified by, 260  
ATM over PON (APON), 229  
ATM over SONET, 90–91  
attenuation coefficients  
coaxial cable, 549  
fiber, 547  
attenuation of fiber, 546–550  
ATUs (ADSL Terminal Units), 235

- audio streams, 42–43
- authentication, 641
- authentication header, 177
- Automatic Repeat Request (ARQ) protocol, 326, 368
- autonomous systems (ASs), 172–173
- available bit rate (ABR)
  - ATM, 260–262
  - GCRA, 395, 397–398
- averaging rate fluctuations, 407–408
  
- B**
  
- B channels in ISDN, 9, 233–234
- B frames in MPEG standard, 251
- B-ICI (BISDN Inter Carrier Interface), 271
- backbone networks
  - digital, 205–206
  - Internet, 157–158
  - in network resource models, 505
  - telephone network, 50
- backward compatibility
  - Internet, 184
  - SONET, 212
- backward explicit congestion notification (BECN), 140–141
- Bahadur-Rao theorem, 410, 480–482
- balanced equations, 435
- bands in FDM, 62
- bandwidth
  - ATM networks, 476–478
  - as bottleneck, 91–92
  - CATV systems, 245
  - circuit-switched networks, 209
  - digitization, 22
  - FDM, 62–63
  - fiber, 547–548
  - local loops, 206
  - in QoS, 370–371
  - wireless networks, 320–321
- bandwidth balancing, 138
- bandwidth-distance product, 550–552
- banyan networks, 595–596
- base stations, 308, 344–345
- Batcher networks, 603–605
- BCH (Bose-Chaudhury-Hocquenghem) code, 70
- bearer services, 40
  - ATM as, 259
  - circuit-switched networks, 209
  - ISDN, 232–233
  - in ODN model, 86–88
- BECN (backward explicit congestion notification), 140–141
- beginning of message (BOM) segments, 144
- Bell, Alexander Graham, 6
- Bellman-Ford algorithm
  - datagram networks, 384–385
  - IP routing, 169–170
  - multicast IP, 174
- Benes networks, 585–587
- BER (bit error rates)
  - acceptable, 45–46
  - optical links, 543, 545
  - wireless networks, 319
- best-effort services, 160, 185, 370, 510
- BGP (Border Gateway Protocol), 172–173
- BGWs (billing gateways), 515–516
- bidirectional networks, 19
- billing and provisioning systems for Internet connections, 511–512
- INDEX, 515–518
- user experience in, 512–513
- variable quality in, 513–515
- billing domains, 517–518
- billing gateways (BGWs), 515–516
- binary representation, 21–22
- BISDN Inter Carrier Interface (B-ICI), 271

- BISDN reference model, 285–286, 293–294  
bit error rates (BER)  
    acceptable, 45–46  
    optical links, 543, 545  
    wireless networks, 319  
bit-level transparency, 557  
bit rates  
    network resource models, 505  
    optical links, 543  
bit streams, 7  
bit times in multiplexing, 59  
bit ways layer in ODN model, 86–88  
block mode in FTP, 182  
blocking and blocking probability  
    circuit-switched networks, 208,  
        372–374, 446–448  
    datagram networks, 462–465  
    dynamic wavelength assignment,  
        568–569  
    input buffers, 611–614  
    insensitivity of, 462–465  
    Markov chain model for, 444–445  
Bluetooth wireless networks, 358  
BOM (beginning of message)  
    segments, 144  
Border Gateway Protocol (BGP),  
    172–173  
border gateway speakers, 172–173  
border gateways, 172  
Bose-Chaudhury-Hocquenghem  
    (BCH) code, 70  
bottlenecks, 50, 91–92  
bridges  
    data link layer, 108–109  
    Ethernet networks, 121–125  
    SMDS, 145–147  
Broadband Integrated Services Digital  
    Network (BISDN), 285–286,  
        293–294  
broadcast domains  
    Ethernet networks, 121  
virtual LANs, 127  
Broadcast/Unknown Server (BUS),  
    295–296  
BT (burst tolerance)  
    ATM, 260, 262  
    GCRA, 397  
budget power of fiber, 549  
buffers  
    ATM networks, 413–419  
    Ethernet switches, 126  
    Frame Relay, 141  
    MMF source, 459–462  
    overflow, 78, 141  
    packet switching, 65  
        distributed, 593–605  
    input, 610–615  
    output, 608–610  
    shared, 605–608  
queuing network model, 52  
SM, 60, 416–419  
SMDS, 144–145  
streaming applications, 42  
TCP, 178  
TDM, 60  
burst tolerance (BT)  
    ATM, 260, 262  
    GCRA, 397  
burstiness, 468–469  
BUS (Broadcast/Unknown Server),  
    295–296  
buses, 84–85  
business commerce, 3  
busy channels, 59  
byte interleaving, 212  
byte-stream services, 178
- C —
- cable modem termination system  
    (CMTS), 248  
cable modems  
    CATV, 19, 247–250

- cable modems (*cont.*)  
     passive optical networks for, 227
- cable TV. *See* CATV (Community Antenna Television) systems
- CAC (call admissions control), 268
- caches, 196
- call admission, 208
- call admissions control (CAC), 268
- call forwarding service, 240–241
- call handoffs, 346, 632–635
- call phase, 64
- Call Proceeding acknowledgments, 272
- calls  
     circuit-switched networks, 64, 208  
     pricing, 403–405  
     usage charges, 507  
     wireless networks, 339–340
- CAM (contents-addressable memory), 592
- CAP (carrierless amplitude-modulation-phase) modulation, 236
- capacity of wireless networks, 323–324
- carried calls, 211
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD), 118
- carrier sensing, 335
- carrierless amplitude-modulation-phase (CAP) modulation, 236
- CATV (Community Antenna Television) systems, 17, 244–245  
     future, 29–30  
     history, 17–19  
     layered network, 247–248  
     layout, 245–247  
     MPEG in, 250–252  
     SCM for, 555  
     services over, 249–250
- wireless, 246–247
- CBR (constant bit rate), 45–46  
     ATM, 260–262  
     GCRA, 395–396
- CC (connection count) numbers, 194
- CCS (common channel signaling)  
     in INA model, 242  
     telephone networks, 7
- cd command in FTP, 182
- CD (count-down) counters, 137
- CDMA (code-division multiple access) technique, 332–333
- CDPD (cellular digital packet data) systems, 351
- CDs (compact discs), 23
- CDV (cell delay variation), 261–262
- CDVT (cell delay variation tolerance)  
     ATM, 260, 262  
     GCRA, 396
- cell delay variation (CDV), 261–262
- cell error ratio (CER), 261
- cell loss priority (CLP), 278, 280–281
- cell loss ratio (CLR), 261–262
- cells  
     ATM, 16, 262–266  
     cellular telephone systems, 344  
     mobile telephone service, 19–20
- cellular digital packet data (CDPD) systems, 351
- cellular networks, 307–308, 344–348, 357
- central offices, 57
- CER (cell error ratio), 261
- channel IDs (CIDs), 284
- channel inversion, 328–329
- channels  
     ADSL, 235  
     ISDN, 9–10, 233–234  
     in multiplexing, 59, 61  
     wireless networks, 315–316, 331–332
- coding, 325–326

- multiple access, 332–334
- random access, 334–337
- spectral etiquette, 337
- checksum code (CKS), 68
- checksums in TCP, 179
- CIDR (classless interdomain routing), 165
- CIDs (channel IDs), 284
- CIR (committed information rate), 141–142
- circuit-switched networks, 63–65, 205–207
  - blocking probability, 446–448
- CATV, 244–245
  - layered network, 247–248
  - layout, 245–247
  - MPEG in, 250–252
  - services over, 249–250
  - wireless, 246–247
- complexity considerations, 449
- control of, 372
  - blocking in, 372–374
  - network, 446–450
  - routing optimization, 374–378
  - single switch, 443–446
- digital subscriber line, 232
  - ADSL, 235–239
  - ISDN, 232–235
- DWDM, 223–225
- Erlang fixed point in, 449–450
- Erlang loss formula for, 446
- fiber to the home, 225
  - hybrid schemes, 231–232
  - optical loop carrier system, 225–226
- passive optical networks in, 226–230
- passive photonic loops in, 230–231
- insensitivity in, 446
- intelligent, 239
  - functional components, 243–244
- INA model, 241–243
- service examples, 239–241
- invariant distributions in, 445
- Markov chain model for, 444–445
  - vs. packet-switched, 588–589
- performance, 208–211
- SONET, 211–215
  - frame structure, 215–221
  - optical networking and future of, 222–223
  - telephone networks, 6
- circuits, 64, 208
- CKS (checksum code), 68
- class-based IP addresses, 164
- class-based queuing, 191
- classless interdomain routing (CIDR), 165
- clear channels, 568
- CLECs (competitive local exchange carriers), 237–238
- client/server applications, 43–44
- clocks in SONET, 211–212
- Clos networks, 582–588
- closed-loop applications, 629
- CLP (cell loss priority), 278, 280–281
- CLR (cell loss ratio), 261–262
- ClusterControlVC, 300
- clusters, 300
- CM (continuing message) segments, 144
- CMTS (cable modem termination system), 248
- coaxial cable
  - attenuation coefficient, 549
  - CATV systems, 17–18, 244–245
- code-division multiple access (CDMA) technique, 332–333
- code-division multiplexing, 63
- codecs, 18–19
- codewords in error correction, 70

- coding
  - global multimedia networks, 627, 638
  - video, 631–632, 637–638
- wireless networks
  - channels, 325–326
  - for flat-fading, 327–328
- coherence bandwidth, 320–321
- coherence in lasers, 544
- coherent detection, 325
- collisions
  - ALOHA technique for, 335
  - DQDB, 137–138
  - Ethernet networks, 118–121
  - optical LANs, 562
- com domain, 166
- committed information rate (CIR), 141–142
- common channel signaling (CCS)
  - in INA model, 242
  - telephone networks, 7
- Community Antenna Television. *See* CATV (Community Antenna Television) systems
- compact discs (CDs), 23
- compatibility
  - cellular phone standards, 308
  - Internet, 184
  - SONET, 212
- competitive local exchange carriers (CLECs), 237–238
- complexity
  - circuit-switched networks, 449
  - global multimedia networks, 639–641
  - switches, 576–577
- compression
  - CATV, 18–19
  - FTP, 182
  - global multimedia networks, 625
  - MPEG, 250–252
  - in OSI model, 113
- video, 631–632, 637–638
- computer networks, history, 10–17
- confidentiality, 641
- configurability, 637–639
- configuration
  - ATM network status, 291–292
  - switch, 578–579
- conformant cells, 395–396
- congested states, 376
- congestion and congestion control, 58, 78, 366
  - ATM, 290
  - datagram networks, 387–388
    - rate, 391–392
    - window, 388–391
  - Frame Relay, 141
  - IP protocols, 159
  - reservation protocols, 336
- congestion charges, 506–508, 523–526
- Connect messages, 272
- CONNECT service, 111
- connection count (CC) numbers, 194
- connection-oriented services, 48
  - ATM as, 259–262
  - in transport layer, 110–111
- connection requests, 243–244
- connection setup
  - circuit switching, 64
  - PNNI, 274
- connection teardown, 64
- connectionless services, 48–49, 110–111
- connections
  - ATM, 271, 274
  - circuit-switched networks, 64, 208
  - intelligent networks, 243–244
    - usage charges, 507
  - connectivity of switches, 576
  - connectors in ISDN, 234
- constant bit rate (CBR), 45–46
  - ATM, 260–262
  - GCRA, 395–396

- consumer surplus, 498  
contents-addressable memory (CAM), 592  
continuing message (CM) segments, 144  
continuous-time Markov chains, 438–443  
control functions in INA model, 242  
control messages  
  CATV, 19  
  in layer implementation, 83  
control of networks, 363–365  
  ATM networks, 285–287, 392–393, 465–466  
  Bahadur-Rao theorem, 480–482  
  control problems, 393–395  
  deterministic approaches, 395–405, 466–469  
  deterministic vs. statistical procedures, 422–424  
  fault management, 287–290  
  large variations of random variables iid, 470–474  
  network status monitoring and configuration, 291–292  
  queue large deviations, 475–480  
  statistical procedures, 405–422  
  straight-line large deviations, 474–475  
  traffic and congestion control, 290  
  user/network signaling, 292  
circuit-switched networks, 372  
  blocking in, 372–374  
  network, 446–450  
  routing optimization, 374–378  
  single switch, 443–446  
datagram networks  
  blocking probability, 462–465  
  buffer occupancy for MMF source, 459–462  
  congestion control, 387–392  
discrete-time queues, 453–456  
Jackson networks, 456–459  
key queuing result, 379–381  
M/M/1 queues, 450–452  
queuing model, 378–379  
routing optimization, 381–387  
examples, 368–369  
Markov chains, 431–432  
  continuous-time, 438–443  
  discrete time, 432–438  
methods, 365–367  
QoS in, 369–372  
  time scales in, 367–368  
control planes, 286  
controlled-load service, 195  
conversion sublayer (CS), 282–283  
cordless phones, 348–349  
corruption, 628  
costs. *See also* economics; pricing communication, 491  
economies of scale, 24–25, 56–57, 184  
information goods, 493  
for ISPs, 496–497  
single resource pricing, 526–527  
count-down (CD) counters, 137  
count-down timers  
  multicast IP, 173  
  packet transmissions, 70  
counters in DQDB, 137  
coupler losses, 548–549  
CPCS sublayer, 283  
CPUs in layer implementation, 84–85  
Cramer's theorem, 470–474  
crankback procedure, 272  
CRC (cyclic redundancy check)  
  ATM, 262, 281  
  in data link layer, 105–106  
  Ethernet packets, 118  
  operation, 68–70  
SDLC, 11

- CRC (cyclic redundancy check) (*cont.*)  
     SMDS, 143–144
- critical size  
     global multimedia networks, 640–641  
     importance of, 25–26
- cross-talk  
     FDM, 62  
     serial transmissions, 10
- crossbar switches, 577, 610–611
- crosspoints, 577–579, 596
- CS (conversion sublayer), 282–283
- CSMA/CD (Carrier Sense Multiple Access with Collision Detection), 118
- cut-through forwarding, 126
- cyclic redundancy check. *See* CRC (cyclic redundancy check)
- 
- D**
- D channels, 9–10, 233–234
- DA (destination address)  
     Ethernet packets, 118  
     IPv6, 176–177  
     packet switching, 67, 589–590
- DAB (Digital Audio Broadcasting) system, 354
- dark current, 545
- data link connection identifiers (DLCIs), 140–141
- data link layer in OSI model, 105–106  
     logical link control, 107–109  
     media access control, 106–107
- Data-Over-Cable Service Interface Specifications (DOCSIS), 248–249
- datagram networks  
     control of  
         blocking probability, 462–465  
         buffer occupancy for MMF source, 459–462
- congestion control, 387–392
- discrete-time queues, 453–456
- Jackson networks, 456–459
- key queuing result, 379–381
- M/M/1 queues, 450–452
- queuing model, 378–379
- routing optimization, 381–387  
     routing in, 65–67
- dB (decibels), 23
- DBS (distributed buffer switch) designs, 593
- DCS (digital cross-connect system), 215
- DCT (digital cosine transform) compression scheme, 250
- DD (depacketization delay), 263, 265
- DE (discard eligibility), 140–141
- decapsulation  
     in data link layer, 106  
     mobile IP, 175–176
- decibels (dB), 23
- decision-feedback equalizers (DFEs), 329
- decision operation in optical receivers, 544–545
- decomposition, layers in, 81
- decoupling bandwidth, 416, 480
- DECT cordless phone systems, 349
- dedicated access links, 208
- dedicated resources, 624
- deep fade condition, 318
- delay agents, 640
- delays  
     analysis, 53–56  
     ATM networks, 263–265, 393–394  
     circuit-switched networks, 65, 209  
     congestion control, 387–392  
     connectionless services, 49  
     discrete-time queues, 453–456  
     distributed-gradient algorithm, 385–387  
     Ethernet networks, 118–121

- global multimedia networks, 627–628, 637  
high-performance networks, 51  
Jackson networks, 456–459  
key queuing result for, 379–381  
 $M/GI/\infty$  queues, 454–456  
 $M/M/1$  queues, 451–452  
packet switching, 65, 67  
in QoS, 370–371  
satellite networks, 353  
static routing, 382–383  
streaming applications, 42  
switches, 576, 581  
wireless networks, 320  
World Wide Web, 42  
delta networks, 593–594  
demand  
    derived, 491–492  
        information goods, 492–493  
        site rents, 493–494  
    for ISPs  
        diversity in, 502–503  
        model, 497–501  
demand-assignment protocols, 334–336  
demand curves, 497  
demultiplexing, 59  
    FDM, 62  
    SM, 62  
    TDM, 580–581  
dense wave-division multiplexing (DWDM), 223–225  
depacketization delay (DD), 263, 265  
derived demand, 491–492  
    information goods, 492–493  
    site rents, 493–494  
design of wireless networks  
    architecture, 337–339  
    internetworking, 341  
    mobility management, 339–340  
    new paradigm, 342–343  
    reliability, 340–341  
    security, 341–342  
designated receivers, 175  
Designated Transit Lists (DTLs), 273  
destination addresses  
    Ethernet packets, 118  
    IPv6, 176–177  
    packet switching, 67, 589–590  
destination options header, 177  
destination port numbers, 178–179  
destination service access points (DSAPs), 118  
destinations, 51, 169–170  
deterministic approaches to ATM networks, 395  
admission control, 398–403  
ATM Forum recommendations, 395–398  
for burstiness, 468–469  
leaky bucket, 467–468  
linear bounds, 466–467  
pricing calls, 403–405  
vs. statistical, 422–424  
DFEs (decision-feedback equalizers), 329  
DHCP (Dynamic Host Configuration Protocol), 165  
differential detection, 325  
Differential Pulse Code Modulation (DPCM), 250  
Digital Audio Broadcasting (DAB) system, 354  
digital backbone networks, 205–206  
digital carrier systems, 7  
digital cordless phones, 348  
digital cosine transform (DCT) compression scheme, 250  
digital cross-connect system (DCS), 215  
digital loop carrier systems, 225  
digital modulation techniques, 324–325

- digital subscriber line (DSL), 232  
    ADSL, 235–239  
    ISDN, 232–235  
Digital Subscriber Loop Access Multiplexer (DSLAM), 237  
digital technology  
    CATV, 18–19  
    cellular networks, 346–347  
    telephone networks, 7  
digital transparency, 557  
digitization, 21–24  
Dijkstra's algorithm, 171–172, 273  
direct addressing, 592  
direct sequence spread spectrum technique, 330  
directional antennas, 331  
directive infrared transmissions, 322  
directory servers, 166  
disaggregation of QoS, 630  
discard eligibility (DE), 140–141  
DISCONNECT service, 111  
discover packets, 165  
Discrete Multitone (DMT)  
    modulation, 236  
discrete time in Markov chains, 432–438  
discrete-time queues, 453–456  
dispersion  
    buffering, 415  
    fiber, 550–554  
distance vector algorithms, 170  
distributed buffer switch (DBS)  
    designs, 593  
distributed buffers, 593–598  
    hotspots  
        combating, 601–605  
        impact, 596–597  
    input buffers in, 598–601  
    multicasting in, 605  
distributed-gradient algorithm, 385–387  
Distributed Queue Dual Bus (DQDB), 135–138  
distributed routing, 356  
diversity  
    in demand, 502–503  
    for flat-fading, 327  
DIX Ethernet standard, 118  
DLCIs (data link connection identifiers), 140–141  
DMT (Discrete Multitone) modulation, 236  
DNS (Domain Name System), 165–166  
DOCSIS (Data-Over-Cable Service Interface Specifications), 248–249  
Domain Name System (DNS), 165–166  
domains  
    Ethernet networks, 121  
    INDEX system, 517–518  
    IP protocol, 166  
    virtual LANs, 127  
Doppler frequency shift, 321  
DPCM (Differential Pulse Code Modulation), 250  
DQDB (Distributed Queue Dual Bus), 135–138  
Drop Party messages, 272  
DS signals, 7–8  
DSAPs (destination service access points), 118  
DSL (digital subscriber line), 232  
    ADSL, 235–239  
    ISDN, 232–235  
DSL-lite system, 237  
DSLAM (Digital Subscriber Loop Access Multiplexer), 237  
DTLs (Designated Transit Lists), 273  
DWDM (dense wave-division multiplexing), 223–225  
Dynamic Host Configuration Protocol (DHCP), 165

- dynamic routing  
  circuit-switched networks, 375  
  datagram networks, 383–384
- dynamic tables, 590
- dynamic wavelength assignment, 568–569
- E —
- EA information, 140
- economics, 489–491. *See also* costs; pricing
- ATM services, 528
- resources and services model, 529–533
- revenue maximization, 533–535
- billing and provisioning systems  
  for Internet connections, 511–512
- INDEX, 515–518
- user experience in, 512–513
- variable quality in, 513–515
- changes, 3–4
- derived demand, 491–494
- information goods, 492–493
- ISPs, 494–497  
  empirical evidence, 501–504  
  subscriber demand model, 497–501
- network charges, 504–505  
  economic principles, 506–509  
  and Internet vulnerability, 510–511  
  in practice, 509  
  resource model, 505–506
- single resource pricing, 518–519  
  congestion prices, 523–526  
  cost recovery and optimum link capacity, 526–527  
  usage-based, 520–523
- site rents, 493–494
- economies of scale, 24–25
- Internet, 184
- transmissions, 56–57
- EDFA (Erbium doped fiber amplifiers), 549–550
- edu domain, 166
- effective bandwidth, 476–478
- efficiency  
  ABR, 74
- Ethernet networks, 120
- FDM, 63
- Frame Relay, 140
- Go Back N, 75
- power plants, 24–25
- TFTP, 183
- token ring networks, 129–130
- encapsulating security payload  
  header, 177
- encapsulation  
  data link layer, 106
- mobile IP, 175–176
- encapsulation over AAL5, 295
- encoding  
  FDDI, 132
- global multimedia networks, 625, 627
- 10BASE-T networks, 115–116
- encryption  
  global multimedia networks, 625, 627, 636–637, 641
- in OSI model, 113
- end of message (EOM) segments, 144
- end-to-end control  
  in data link layer, 106
- Frame Relay, 139
- end-to-end delays, 209
- ending delimiters, 128
- EOM (end of message) segments, 144
- equal access guarantee, 25
- equalizer techniques, 329
- Erbium doped fiber amplifiers (EDFA), 549–550

Erlang fixed point, 449–450  
 Erlang loss formula, 446  
 error bursts, 320  
 errors and error handling, 58,  
     68–72  
 AAL, 284  
 ABP for, 72–75  
 ATM, 262, 281  
 CATV systems, 248  
 in data link layer, 105–108  
 digitization, 22–23  
 FDDI, 133  
 Frame Relay, 138–139  
 Go Back N, 75–77  
 IP protocols, 159  
 IPv6, 177  
 in physical layer, 105  
 SDLC, 11–12  
 SMDS, 143–144  
 SRP, 77  
 TCP, 179, 189  
 X.25 protocol, 138  
 Ethernet networks, 13–15, 114–115  
     bridges, 121–125  
     introduction of, 307  
     LAN interconnections, 122–127  
     LLC, 121–122  
     MAC, 118–121  
     physical layer, 115–118  
     switches, 121, 125–127  
     virtual LANs, 127  
 etiquette for wireless networks,  
     337  
 EXP\_DATA service, 111  
 explicit congestion notification, 189,  
     391  
 exponentially distributed random  
     variables, 438  
 extensibility, 621, 639–641  
 extension headers, 176  
 exterior routers, 172  
 externalities, 25–26, 156

**F**

fading  
     global multimedia networks, 634  
     wireless networks, 317–321  
 fair allocation, 135  
 fair queuing, 191  
 far end receive failure (FERF) field,  
     289  
 fault management, 287–290  
 FCC (Federal Communications  
     Commission), 24  
 FCFS (first-come, first-served) basis  
     DQDB, 135–136  
     limitations, 190–191  
 FD (fixed processing delay), 263, 265  
 FDDI (Fiber Distributed Data  
     Interface), 15–16, 131–135  
 FDM (frequency-division  
     multiplexing)  
     CATV systems, 244  
     operation, 59, 62–63  
 FDMA (frequency-division multiple  
     access) technique, 332–333  
 FECN (forward explicit congestion  
     notification), 140–141  
 Federal Communications Commission  
     (FCC), 24  
 FERF (far end receive failure) field,  
     289  
 fiber. *See also* optical networks  
     attenuation, 546–550  
     CATV systems, 17–18, 245  
     dispersion, 550–554  
     100BASE-T networks, 117  
 Fiber Distributed Data Interface  
     (FDDI), 15–16, 131–135  
 fiber in the loop (FTL), 225  
 fiber to the curb (FTTC), 18, 555  
 fiber to the home (FTTH), 225  
     hybrid schemes, 231–232  
     optical loop carrier system, 225–226

- passive optical networks in, 226–230  
passive photonic loops in, 230–231  
FIFO (first in, first out) buffers, 60  
File Transfer Protocol (FTP), 181–182  
filters  
    for intersymbol interference, 329  
optical receivers, 544–545  
wireless networks, 321–322  
FIN flag, 179–180  
first-come, first-served (FCFS) basis  
    DQDB, 135–136  
    limitations, 190–191  
first in, first out (FIFO) buffers, 60  
FITL (fiber in the loop), 225  
fixed cell size, 262–266  
fixed frames, 562  
fixed processing delay (FD), 263, 265  
fixed wireless access, 357–358  
flat-fading  
    countermeasures, 326–328  
    wireless networks, 318–321  
flat-rate charges, 498–502  
flexibility  
    global multimedia networks, 640  
    INDEX pricing, 517–518  
flow control, 58, 77–78  
    Frame Relay, 141  
    IP protocols, 159  
flow RED mechanism, 391  
flows in IP packets, 571  
forced emissions, 544  
foreign agents, 175  
forward explicit congestion  
    notification (FECN), 140–141  
4B/5B encoding, 132  
fragmentation  
    IP protocol, 167–168  
    IPv6 protocol, 177  
frame-based user-network interface  
    (FUNI), 271  
Frame Relay over ATM, 301  
Frame Relay protocols, 138–142, 509  
frames  
    circuit-switched networks, 208  
    Ethernet networks, 118  
    SONET, 215–221  
    TDM, 59  
framing patterns, 59  
free-space propagation model,  
    316–317  
frequency-division multiple access  
    (FDMA) technique, 332–333  
frequency-division multiplexing  
    (FDM)  
        CATV systems, 244  
        operation, 59, 62–63  
frequency hopping spread spectrum  
    technique, 330  
frequency reuse  
    cellular telephone systems, 344–345  
    wireless networks, 321  
frequency-selective fading, 320  
frequency-selective switches, 557  
frequency shift, 321  
FT (function type) field, 289  
FTP (File Transfer Protocol), 181–182  
ftp scheme in URLs, 167  
FTTC (fiber to the curb), 18, 555  
FTTH (fiber to the home), 225  
    hybrid schemes, 231–232  
    optical loop carrier system, 225–226  
    passive optical networks in, 226–230  
    passive photonic loops in, 230–231  
full-duplex transmissions, 117  
function type (FT) field, 289  
functional components in INA model,  
    242–244  
FUNI (frame-based user-network  
    interface), 271  
future networks, 27  
    ATM, 29  
    CATV, 29–30  
    Internet, 27–29  
    predictions, 30–32

future networks (*cont.*)  
 SONET, 222–223  
 wireless, 30, 309–312

## — G —

games, networked, 43  
 gateway-centered approach, 635  
 gateways, 169, 172  
 gating, optical, 559  
 GCRA (generalized cell rate algorithm), 260, 395–398, 419–421  
 generic flow control (GFC), 277, 280  
 geosynchronous (GEO) satellite networks, 309, 353  
 get command in FTP, 182  
 GFC (generic flow control), 277, 280  
 GFR (guaranteed frame rate) services, 260–261  
 global multimedia networks, 619–620  
   applications, 626  
   architectures, 623–624, 627–628  
   attributes, 620–622  
   challenges, 626  
   extensibility and complexity  
     management, 639–641  
   heterogeneity, 636–637  
   mobility, 632–635  
   networking, 624–625  
   QoS, 621, 624–625, 628–632  
   scalability and configurability, 637–639  
   security, 621, 625, 627, 636–637, 641–642  
   signal processing, 625  
   technology areas, 622–623  
 Global Positioning System (GPS), 354  
 GlobalStar satellite system, 353  
 Go Back N protocol, 75–77  
   Ethernet networks, 122  
   Frame Relay, 139–140

TCP, 178, 180  
 X.25, 138  
 gopher scheme in URLs, 167  
 gov domain, 166  
 government regulation, 24  
 GPS (Global Positioning System), 354  
 GRIN (graded-index) profile, 552–553  
 Group Address, 270  
 group addresses  
   ATM, 270  
   multicast IP, 173  
 guaranteed delays, 65  
 guaranteed frame rate (GFR) services, 260–261  
 guaranteed service, 195  
 guard bands  
   FDM, 62  
   TPON systems, 229

## — H —

H channels, 233–234  
 handoffs, 346, 632–635  
 handshaking, 178–179  
 hard capacity, 333  
 hashing, 592  
 HDLC (High-Level Data Link Control), 11, 368  
 HDSL (high data rate DSL), 237  
 head of line (HOL) blocking, 611–614  
 head stations in CATV systems, 245  
 header error-control (HEC), 262–263, 278, 281  
 headers  
   ATM, 277–280  
   other fields in, 280–281  
   reserved VCI/VPI in, 281–282  
   VCIs and VPIs in, 278–280  
 Frame Relay, 140  
 IP packets, 163  
 IPv6, 176–178  
 packets, 11–12

- SMDS, 144  
TCP, 178–179  
HEC (header error-control), 262–263, 278, 281  
heterogeneity, 621, 636–637  
HFC (hybrid fiber/coaxial) systems, 18, 244  
hidden terminal problem, 335  
hierarchical architectures, 338  
hierarchical mesh networks, 570–571  
high data rate DSL (HDSL), 237  
High-Level Data Link Control (HDLC), 11, 368  
high-performance networks  
  performance, 50–51  
  traffic increase, 49–50  
high-speed digital cellular networks, 357  
HIPERLAN family, 354–355  
history, 5–6  
  CATV networks, 17–19  
  computer networks, 10–17  
  telephone networks, 6–10  
  wireless networks, 19–21, 306–309  
HOL (head of line) blocking, 611–614  
holding time, 64, 210  
home agents, 175–176  
home IP addresses, 175–176  
HomeRF wireless networks, 358  
hop-by-hop options header, 177  
horizontal handoffs, 634  
host numbers, 164  
hosts  
  IP, 159  
  multicast IP, 173–174  
hotspots in distributed buffers  
  combating, 601–605  
  impact, 596–597  
HTTP (Hypertext Transfer Protocol), 183, 496  
HTTP/1.1 protocol, 194–195  
http scheme in URLs, 167  
hubs  
  Ethernet networks, 121  
  10BASE-T networks, 115–116  
hybrid fiber/coaxial (HFC) systems, 18, 244  
hybrid schemes in FTTH, 231–232  
Hypertext Transfer Protocol (HTTP), 183, 496
- I —
- ICANN (Internet Corporation for Assigned Names and Numbers), 166  
ICMP (Internet Control Message Protocol), 168–169  
ICR (initial cell rate), 260  
idle channels, 59  
IEEE 802.3 standards. *See* Ethernet networks  
IEEE 802.5 standards. *See* token ring networks  
IGMP (Internet Group Management Protocol), 173  
iid random variables, 470–474  
ILMI (Integrated Local Management Interface) protocol, 269–270, 291–292  
IMEs (Interface Management Entities), 291–292  
IMT-2000 standard, 356–357  
INA (Intelligent Network Architecture) model, 241–243  
INDEX (Internet Demand Experiment)  
  demand diversity results, 502–504  
  flexibility, 517–518  
  operation, 515–517  
  user experience, 512–513  
  variable quality, 513–515  
indium gallium arsenide (InGaAs) diodes, 550

- Industrial, Scientific, and Medical (ISM) frequency bands, 307  
information goods, 492–493  
information workers, 3  
infrared wireless networks, 322–323  
InGaAs (indium gallium arsenide) diodes, 550  
initial cell rate (ICR), 260  
input buffered switches, 605  
input buffers  
    distributed buffer switches, 598–601  
    packet switching, 610–611  
        HOL blocking, 611–614  
        multicasting, 614–615  
        queuing network model, 52  
insensitivity  
    blocking probability, 462–465  
    circuit-switched networks, 446  
int domain, 166  
integrated adaptive protocol design, 342–343  
Integrated Local Management Interface (ILMI) protocol, 269–270, 291–292  
integrated model for IP over ATM, 299  
Integrated Services Digital Network (ISDN), 9–10, 232–235  
Integrated Services (IS) model, 195  
integration, service, 26–27  
integrity, 641  
Intelligent Network Architecture (INA) model, 241–243  
intelligent networks (INs), 239  
    functional components, 243–244  
    INA model, 241–243  
    service examples, 239–241  
intelligent peripherals (IPs), 242  
interactive services, 244–245  
interarrival time, 210–211  
intercell interference, 344  
interconnected services, 40  
Interface Management Entities (IMEs), 291–292  
interference, 318–322, 344, 347, 350  
interference-limited design, 347  
interior routers, 172  
interleaving, 212, 327–328  
International Mobil Telecommunications 2000 standard, 356–357  
Internet, 155–158  
    billing and provisioning systems, 511–512  
    INDEX, 515–518  
    user experience in, 512–513  
    variable quality in, 513–515  
    future of, 27–29  
    successes and limitations, 183–186  
    summary, 196–197  
Internet Control Message Protocol (ICMP), 168–169  
Internet Corporation for Assigned Names and Numbers (ICANN), 166  
Internet Group Management Protocol (IGMP), 173  
Internet Protocol (IP), 13, 158–163  
    addressing, 163–167  
    fragmentation/reassembly, 167–168  
    improvement suggestions, 190  
    IPv6, 176–178  
    mobile, 175–176  
    multicast IP, 173–174  
    optical networks, 571  
    reliable multicast, 174–175  
    routing, 167–173  
    subnets, 159  
    switching, 192  
Internet service providers. *See* ISPs  
    (Internet service providers)  
internetworking  
    ATM, 294–295

- Frame Relay and SMDS over ATM, 301  
IP over ATM, 297–300  
LAN emulation over ATM, 295–296  
multiprotocol over AAL5, 295  
multiprotocol over ATM, 300–301  
protocols, 158  
SMDS, 145–147  
wireless networks, 341  
interoperability, 637  
interpolation frames, 251  
intersymbol interference (ISI)  
countermeasures, 328–331  
wireless networks, 318–321  
invariant distributions  
circuit-switched networks, 445  
Markov chains, 434–436  
IP. *See* Internet Protocol (IP)  
IP addresses, 164–167  
IP models, 104  
IP Multicast Backbone network, 174  
IP over ATM, 297–300, 571  
IPs (intelligent peripherals), 242  
IPv6, 176–178  
Iridium satellite system, 353  
irreducible rate matrices, 441  
irreducible transition probability matrices, 435  
IS (Integrated Services) model, 195  
ISDN (Integrated Services Digital Network), 9–10, 232–235  
ISI (intersymbol interference)  
countermeasures, 328–331  
wireless networks, 318–321  
ISM (Industrial, Scientific, and Medical) frequency bands, 307  
ISPs (Internet service providers)  
demand diversity, 502–503  
empirical evidence, 501–504  
flat-rate charges, 498–502  
price sensitivity, 503–504  
subscriber demand model, 497–501
- ■ ■ J
- Jackson networks, 456–459  
jitter  
ATM, 263, 265  
SONET, 212  
join request messages, 173  
joint source/channel coding, 627, 638  
JPEG (Joint Photographic Experts Group), 250
- ■ ■ K
- key queuing result, 379–381  
knockout switches, 609–610
- ■ ■ L
- labels  
in Dijkstra's algorithm, 171–172  
in TCP/IP networks, 191–193  
LAN emulation (LANE) over ATM, 295–296  
LANs  
connected to Internet, 28  
interconnections, 122–127  
optical  
multihop, 563–565  
single-hop, 561–563  
virtual, 127  
wireless, 349–351, 354–355  
LAPB (Link Access Procedure B), 11, 234  
LAPD (Link Access Procedure D), 234–235  
large deviations in ATM networks  
iid random variables, 470–474  
queue, 475–480  
straight-line, 474–475  
laser diodes, 543–544

- lasers, 223  
layer management planes, 286–287  
layered architectures, 80  
    CATV systems, 247–248  
    layer implementation in, 82–86  
    layers in, 81–82  
    ODN model, 86–89  
layout of CATV systems, 245–247  
leaky-bucket scheme  
    ATM, 394, 467–468  
    datagram networks, 392  
    Frame Relay, 142  
    INDEX system, 516–517  
    SMDS, 144–145  
leave request messages, 173  
leaves in ATM, 271  
LEDs (light-emitting diodes), 543–544  
length indicator (LEN) in Ethernet  
    packets, 118  
length indicator (LI) in AAL, 284  
LEO (low-earth orbit) satellite  
    networks, 309, 353  
LI (length indicator) in AAL, 284  
light-emitting diodes (LEDs), 543–544  
lightpaths, 559–560, 565  
line overhead (LOH)  
    BISDN, 293  
    SONET, 216–219  
linear bounds, 466–467  
linear equalizer techniques, 329  
linear modulation techniques,  
    324–325  
lines, 216–217  
Link Access Procedure B (LAPB), 11,  
    234  
Link Access Procedure D (LAPD),  
    234–235  
link layer retransmission, 325–326  
link level design for wireless networks,  
    324  
    channel coding and link layer  
        retransmission, 325–326  
for flat-fading, 326–328  
for intersymbol interference,  
    328–331  
modulation techniques, 324–325  
Link Management Module MIB  
    module, 291–292  
link state parameters, 273  
links, 51  
    optical. *See* optical networks  
in physical layer, 104–105  
queuing network model, 52  
Web pages, 42  
Little's result, 452  
LLC (logical link control)  
    data link layer, 107–109  
    Ethernet networks, 118, 121–122  
    token ring networks, 130  
LMDS (local multipoint distribution  
systems), 358  
load balancing, 196  
local loops, 206  
local multipoint distribution systems  
    (LMDS), 358  
local packets, 123  
local syntax, 112–113  
location management  
    cellular telephone systems, 346  
    wireless networks, 339–340  
location rents, 493–494  
locations, physical and virtual,  
    491–492  
lock-in effect, 489–490  
logarithmic moment generating  
    function, 409  
logical link control (LLC)  
    data link layer, 107–109  
    Ethernet networks, 118, 121–122  
    token ring networks, 130  
LOH (line overhead)  
    BISDN, 293  
    SONET, 216–219  
look ahead technique, 613–614

loopbacks, 289  
losses and loss rates  
  ATM networks, 393–394  
  global multimedia networks, 624,  
    628  
  in QoS, 370–371  
  queues, 475–480  
low-earth orbit (LEO) satellite  
  networks, 309, 353  
low-pass filters, 544–545

## ■ M

M/GI/infinity queues, 454–456  
M/M/1 queues, 450–452  
MAC (media access control) layer and  
  addresses  
  ATM, 269  
  in data link layer, 106–107  
  Ethernet networks, 118–121  
  FDDI, 132–135  
  IP protocol, 163–164  
  LANE, 297  
  tables of, 123, 126  
  token ring networks, 128–130  
MAC protocol  
  CATV systems, 248  
  DQDB, 135–138  
macrocells, 345  
mailto scheme in URLs, 167  
maintenance costs, 24  
maintenance information, 287–288  
management and control in ATM,  
  285–287  
  fault management, 287–290  
  network status monitoring and  
    configuration, 291–292  
  traffic and congestion control, 290  
  user/network signaling, 292  
Management Information Bases  
  (MIBs), 291–292  
Manchester encoding, 115–116

MANs (metropolitan area networks),  
  135  
marginal costs of information goods,  
  493  
Markov chains, 431–432  
  circuit-switched networks, 444–445  
  continuous-time, 438–443  
  discrete time, 432–438  
  steady-state probabilities, 376  
Markov-modulated fluids (MMF),  
  405–406, 459–462  
MARS (Multicast Address Resolution  
  Server), 300  
master clocks, 211–212  
material dispersion, 553  
MAUs (medium access units), 115  
maximum cell transfer delay, 261  
maximum-likelihood sequence  
  estimation, 329  
maximum segment size (MSS), 178  
Maxwell's equations, 317  
MBone network, 174  
MCNS (Multimedia Cable Network  
  Systems) consortium, 248  
MCR (minimum cell rate), 260, 262  
MCS (Multicast Server), 300  
mean cell transfer delay, 261  
media access control. *See* MAC (media  
  access control) layer and  
  addresses  
media in physical layer, 104  
MediaOne acquisition, 205  
medium access units (MAUs), 115  
memoryless random variables, 439  
merging acknowledgments, 189  
message-handling services, 233  
message identifiers (MIDs), 144  
messages  
  layer implementation, 82–86  
  traffic characteristics, 46–47  
  in transport layer, 110–111  
metastable states, 376

- Metricom architecture, 351  
metropolitan area networks (MANs), 135  
MIBs (Management Information Bases), 291–292  
microcells, 345–346  
MID (multiplexing identifier) field, 284  
middleware layer in ODN model, 86–88  
MIDs (message identifiers), 144  
mil domain, 166  
minimum cell rate (MCR), 260, 262  
minimum mean square error equalizer, 329  
Minitel network, 25  
MMDS (multichannel multipoint distribution services), 358  
MMF (Markov-modulated fluids), 405–406, 459–462  
mobile applications, 626  
Mobile IP (Mobil Internetworking Routing Protocol), 175–176, 339–340  
mobile telephone service, 19–20  
cellular networks, 307–308, 344–348, 357  
introduction of, 308  
mobile telephone switching offices (MTSOs), 308, 339–340, 346  
mobility management  
global multimedia networks, 621, 632–635  
mobile IP, 176  
wireless networks, 339–340  
modal dispersion, 553–554  
modems, 10–11  
ADSL, 235–236, 238–239  
bandwidth, 206  
cable, 19, 227, 247–250  
in network resource models, 505  
modular switch designs, 582–588  
modulation techniques  
global multimedia networks, 623–624, 639  
in physical layer, 105  
wireless networks, 324–325  
MPEG (Motion Pictures Expert Group) standard, 18–19, 250–252  
MPLS (multiprotocol label switching), 192  
MPOA (multiprotocol over ATM), 300–301  
MSOs (Multiple Service Operators), 248  
MSS (maximum segment size), 178  
MTSOs (mobile telephone switching offices), 308, 339–340, 346  
multicarrier modulation, 329–330  
Multicast Address Resolution Server (MARS), 300  
multicast IP, 173–174  
multicast IP over ATM, 300  
Multicast Server (MCS), 300  
multicast trees, 193  
multicasting  
distributed buffers, 605  
input buffers, 614–615  
output buffers, 608  
reliable, 174–175  
shared buffers, 607  
multichannel multipoint distribution services (MMDS), 358  
multiclass case, 419  
multihop optical LANs, 563–565  
Multimedia Cable Network Systems (MCNS) consortium, 248  
multimedia networks. *See* global multimedia networks  
multimode fibers, 553  
multipath components, 316  
multipath delay spread, 320  
multipath flat-fading, 318–321

- multiple access, 13–14  
cellular telephone systems, 347  
wireless networks, 332–334
- Multiple Service Operators (MSOs), 248
- multiplexing  
ADSL, 235  
ATM, 266–267, 408–410, 416–419  
CATV systems, 244  
FTTH, 226  
IP protocols, 159  
operation, 58–63  
optical links, 554–555  
packet switching, 12  
SONET, 211–212  
time-division switching, 580–581  
trunks, 57–58
- multiplexing gain, 61, 409
- multiplexing identifier (MID) field, 284
- multiprotocol label switching (MPLS), 192
- multiprotocol over AAL5, 295
- multiprotocol over ATM (MPOA), 300–301
- multistage switches, 577
- N —
- NAPs (network access points), 156–157, 495
- narrowband interference, 321–322
- NAS (Network Access System), 515
- National Information Infrastructure (NII) frequency band, 355
- National Science Foundation, 156–157
- NBMA (nonbroadcast multiaccess)  
link layer, 298–299
- near-far problem, 333
- negative acknowledgments, 175
- network access points (NAPs), 156–157, 495
- Network Access System (NAS), 515
- network architectures, 89–91
- network bottlenecks, 91–92
- network charges, 504–505  
economic principles, 506–509  
and Internet vulnerability, 510–511  
in practice, 509  
resource model, 505–506
- network elements  
examples, 54–56  
principal, 51–53  
and service characteristics, 53–54
- network information revision requests, 244
- Network Interface Cards (NICs)  
in layer implementation, 86  
MAC addresses, 118
- network large deviations, 478–480
- network layer in OSI model, 109–110
- network mechanisms, 56–58  
congestion control, 78  
error control, 68–77  
flow control, 77–78  
multiplexing, 58–63  
resource allocation, 79–80  
switching, 63–68
- network-network interface (NNI), 270–271, 277
- Network News Transfer Protocol (NNTP), 496–497
- network numbers, 164
- network operating centers (NOCs), 516
- network resource status requests, 244
- network resources in INA model, 242
- network services, 39–41  
applications, 41–44  
connection-oriented, 48  
connectionless, 48–49  
high-performance networks, 49–51  
traffic characterization and QoS in, 44–47

- network status monitoring and configuration, 291–292  
networked games, 43  
networking principles, 21  
    digitization, 21–24  
    economies of scale, 24–25  
    externalities, 25–26  
    service integration, 26–27  
news scheme in URLs, 167  
Next Hop Resolution Protocol (NHRP), 298  
NICs (Network Interface Cards)  
    in layer implementation, 86  
    MAC addresses, 118  
NII (National Information Infrastructure) frequency band, 355  
NNI (network–network interface), 270–271, 277  
NNTP (Network News Transfer Protocol), 496–497  
NOCs (network operating centers), 516  
nodal state parameters, 273  
nodes, 51  
noise  
    digitization, 23  
    photodetectors, 545  
    physical layer, 105  
non-negotiated QoS parameters, 261  
non-wavelength continuity (NWC) number, 566  
nonblocking switches, 582  
nonbroadcast multiaccess (NBMA)  
    link layer, 298–299  
nonconformant cells, 395–396  
nondirective infrared transmissions, 322  
nonrepudiation, 641  
notch filters, 321–322  
NTSC television standard, 19  
numbering packets in transport layer, 111  
NWC (non-wavelength continuity) number, 566  
Nyquist's theorem, 22
- O
- OAM cells, 289–290  
OC-n (Optical Carrier-n) elements, 216  
ODN (Open Data Network) model, 86–89, 184  
OFDM (orthogonal frequency division multiplexing), 329–330  
offer packets, 165  
offset field (OSF), 284  
omega networks, 595  
on-line estimation, 410–413  
on-off keying (OOK) transmission, 542  
100BASE-T networks, 116–117  
100BASEx networks, 117  
OOK (on-off keying) transmission, 542  
Open Data Network (ODN) model, 86–89, 184  
open-loop applications, 629  
open loop congestion control, 78  
open shortest path first (OSPF) algorithm, 171–172  
Optical Carrier-n (OC-n) elements, 216  
optical cross connects (OXC), 224, 557–560, 564  
optical fiber  
    attenuation, 546–550  
    CATV systems, 17–18, 245  
    dispersion, 550–554  
    100BASE-T networks, 117  
optical loop carrier system, 225–226

- optical networks, 541–542  
DWDM, 223–225  
dynamic wavelength assignment  
and blocking, 568–569  
FDDI, 15–16, 131–135  
FTTH, 225  
hybrid schemes, 231–232  
optical loop carrier system,  
225–226  
passive optical networks in,  
226–230  
passive photonic loops in,  
230–231  
hierarchical mesh, 570–571  
LANs  
multihop, 563–565  
single-hop, 561–563  
optical cross-connects, 557–560  
optical links, 542–543  
fiber, 546–554  
receivers, 544–546  
subcarrier multiplexing,  
554–555  
transmitters, 543–544  
protocol stacks, 571–572  
ring networks, 569–570  
SONET, 211–215  
frame structure, 215–221  
optical networking and future,  
222–223  
static wavelength assignment,  
565–568  
WDM systems, 556–557  
optimum link capacity, 526–527  
org domain, 166  
organizational unique identifiers,  
118  
orthogonal frequency division  
multiplexing (OFDM), 329–330  
orthogonal spreading codes,  
332–333  
OSF (offset field), 284
- OSI (Open Systems Interconnection)  
model, 104  
application layer, 113  
data link layer, 105–109  
and Internet protocols, 158  
network layer, 109–110  
physical layer, 104–105  
presentation layer, 112–113  
session layer, 111–112  
transport layer, 110–111  
OSPF (open shortest path first)  
algorithm, 171–172  
output buffers  
packet switching, 608–610  
queuing network model, 52  
output expansion, 614  
overflow, buffer, 78, 141  
overhead  
ATM, 260, 263  
BISDN, 293–294  
reservation protocols, 336  
SONET, 216  
overlay model, 269  
OXC (optical cross connects), 224,  
557–560, 564
- ■ ■ P
- P frames, 251  
packet classifiers, 194  
packet radio, 306–307  
Packet-Reservation Multiple Access  
(PRMA) technique, 336  
packet scheduler link-layer  
mechanism, 194  
packet switching and packet-switched  
networks, 63–68, 103,  
588–591  
buffers  
distributed, 593–605  
input, 610–615  
output, 608–610

- packet switching and packet-switched networks (*cont.*)  
buffers (*cont.*)  
    shared, 605–608  
DQDB, 135–138  
Ethernet. *See* Ethernet networks  
FDDI, 131–135  
Frame Relay, 138–142  
in network resource models, 505  
OSI and IP models for, 104–114  
searching routing tables,  
    591–593  
SMDS, 142–147  
token ring, 127–130  
packetization delay (PD), 263  
packets  
    Ethernet networks, 118  
    SDLC, 11–12  
    statistical multiplexing, 60–62  
    transport layer, 111  
PAD (padding field), 118  
paging systems, 308–309, 352  
parity bits, 68–69, 219  
passband technique, 236  
passive optical networks (PONs),  
    226–230  
passive photonic loops (PPLs),  
    230–231  
passwords in URLs, 167  
path loss, 316–317  
path messages, 193  
path overhead (POH)  
    BISDN, 293–294  
    SONET, 216, 220  
paths  
    Markov chains, 434  
    SONET frames, 216–217  
payload length field, 176  
payload type (PT), 277, 280  
PCR (peak cell rate)  
    ATM, 260–262  
    GCRA, 396  
PCS (Personal Communication Systems), 307–308, 344–348, 357  
PD (packetization delay), 263  
PDUs (protocol data units), 143–145  
peak cell rate (PCR)  
    ATM, 260–262  
    GCRA, 396  
peak-to-peak cell transfer delay, 261  
peer-to-peer architectures, 338  
performance  
    circuit-switched networks, 208–211  
    data networks, 16–17  
    high-performance networks, 50–51  
    switches, 576–579  
TCP/IP networks, 186  
    IP improvement suggestions,  
        190  
    label switching, 191–193  
    queuing algorithms, 190–191  
    RSVP improvement suggestions,  
        193–196  
    TCP improvement suggestions,  
        188–190  
    window adjustments, 186–190  
virtual circuit networks, 258  
wireless networks, 20  
periodic probability transition  
    matrices, 437  
Permanent Virtual Circuits (PVCs),  
    140–141  
Personal Communication Systems  
    (PCS), 307–308, 344–348, 357  
Personal Handyphone System (PHS),  
    349  
phase-shift-keying (PSK), 325  
phone-points, 348–349  
photo detection, 544–545  
photo diodes, 544–545  
photonic layer in SONET, 216  
PHS (Personal Handyphone System),  
    349

- physical layer  
Ethernet networks, 115–118  
FDDI, 132  
OSI model, 104–105  
token ring networks, 128
- Physical Layer Convergence Protocol (PLCP), 143
- physical medium dependent (PMD) layer, 132
- picocells, 345
- Picturephones, 26
- ping service, 169
- plain old telephone service (POTS), 239–240
- plane management, 286–287
- PLCP (Physical Layer Convergence Protocol), 143
- PMD (physical medium dependent) layer, 132
- PNNI (private network-network interface), 271–277
- PNNI Topology State Elements (PTSEs), 273
- PNNI Topology State Packets (PTSPs), 273
- POH (path overhead)  
BISDN, 293–294  
SONET, 216, 220
- point-to-multipoint connections, 271
- point-to-point connections, 271
- Point-to-Point Protocol (PPP), 236–237
- points of presence (PoPs), 156, 494
- policy-based routing, 195–196
- polling, 173
- PONs (passive optical networks), 226–230
- PoPs (points of presence), 156, 494
- portability, 634–635
- ports, 10–11  
in switches, 576–577  
in TCP, 178–179  
in transport layer, 110–111
- positive recurrent Markov chains, 435
- POTS (plain old telephone service), 239–240
- power  
optical links, 544  
optical receivers, 545  
wireless networks, 312, 314, 350
- power plant efficiency, 24–25
- PPLs (passive photonic loops), 230–231
- PPP (Point-to-Point Protocol), 236–237
- preambles  
data link layer, 106  
Ethernet packets, 118  
physical layer, 104
- prefixes for subnets, 165
- presentation layer in OSI model, 112–113
- pretransmission coordination protocols, 562
- price discrimination for information goods, 493
- price sensitivity of ISPs, 503–504
- pricing. *See also* costs; economics  
ATM services, 528  
resources and services model, 529–533  
revenue maximization, 533–535
- calls, 403–405
- network charges, 504–505  
economic principles, 506–509  
and Internet vulnerability, 510–511  
in practice, 509  
resource model, 505–506
- and QoS, 632
- single resource, 518–519  
congestion prices, 523–526  
cost recovery and optimum link capacity, 526–527
- usage-based, 520–523

priorities  
 connection-oriented services, 48  
 statistical multiplexing, 62

priority queuing, 191

private network-network interface (PNNI), 271–277

PRMA (Packet-Reservation Multiple Access) technique, 336

problems and opportunities, 4–5

processing control, 243–244

processing delays, 53

progressive scanning, 250–251

propagation delays, 53  
 global multimedia networks, 627  
 satellite networks, 353

protocol data units (PDUs), 143–145

protocol transparency, 557

protocols, 40–41  
 error detection, 70  
 layer implementation in, 82–86  
 new designs, 92  
 optical networks, 571–572

PSH flag, 179

PSK (phase-shift-keying), 325

PT (payload type), 277, 280

PTSEs (PNNI Topology State Elements), 273

PTSPs (PNNI Topology State Packets), 273

pulse spread, 552

PVCs (Permanent Virtual Circuits), 140–141

---

**Q**


---

QAM (quadrature-amplitude modulation), 324–325

QD (queuing delay), 53, 263–265

QoS (quality of service)  
 aspects, 369–372  
 ATM, 261, 285  
 best-effort services, 510

connection-oriented services, 48

global multimedia networks, 621, 624–625, 628–632

ISPs, 499–500

network services, 44–47

RSPV, 193–194

wireless networks, 340–341

QPSK (quadrature phase-shift keying), 248

quadrature-amplitude modulation (QAM), 324–325

quality charges, 508

quality in billing systems, 513–515

quality of service. *See QoS (quality of service)*

quantization, 21–23

quantization intervals, 22

quantization noise, 23

quantum limit, 554

quasi reversible queues, 451

queues, 53  
 ATM, 266  
 datagram networks  
 discrete-time, 453–456  
 M/M/1, 450–452  
 deviations, 475–480  
 DQDB, 135–138  
 layer implementation, 84  
 shared buffers, 607–608  
 TCP/IP networks, 190–191

queuing delay (QD), 53, 263–265

queuing models, 52, 378–379

---

**R**


---

radio paging systems, 308–309

radio systems, 306–307

radio wireless networks vs. infrared, 322–323

RADSL (rate-adaptive DSL), 237

RAM Mobil Data network, 351

random access, 334–337

- random early drop (RED) mechanism, 391
- rate-adaptive DSL (RADSL), 237
- rate control, 78, 391–392
- rate matrices, 439–441
- Rayleigh scattering, 547
- real-time protocol (RTP), 194
- rearrangeably nonblocking (RNB) switches, 582–585
- reassembly in IP protocol, 167–168
- receivers
  - data link layer, 105–106
  - optical LANs, 561–562
  - optical links, 544–546
  - physical layer, 104
  - reliable multicast, 175
- RED (random early drop) mechanism, 391
- redundancy in SONET, 214–215
- Reed-Solomon (RS) code, 70
- refresh frames, 251
- regenerators, 215–216
- regular rate matrices, 440
- release after reception, 129
- release after transmission, 129
- Release messages, 272
- release packets, 165
- reliability
  - global multimedia networks, 621
  - goals, 47
  - in physical layer, 105
  - wireless networks, 340–341
- reliable multicast, 174–175
- reliable service, 178
- remote login service (*rlogin*), 182–183
- remote sensing applications, 353–354
- remote terminals, 225–226
- reordering in IP protocols, 159
- repeaters
  - 10BASE-T networks, 115–116
  - 10BASE5 networks, 117
  - token ring networks, 128
- REQ (request) counters, 137
- request packets, 165
- resequencing in transport layer, 111
- reservation prices, 525
- reservation schemes
  - optical LANs, 562
  - RSVP, 193–194
  - wireless networks, 334–336
- reserved VCI/VPI, 281–282
- resource allocation, 58
  - ATM, 267–269
  - overview, 79–80
- resource management (RM) cells, 398
- resource model
  - ATM, 529–533
  - network charges, 505–506
- Resource Reservation Protocol (RSVP), 193–196
- resource status requests, 244
- resources in INA model, 242
- resume command, 183
- reuse distance, 344
- revenue maximization, 533–535
- ring networks
  - optical, 569–570
  - token ring, 127–130
- rlogin (remote login service), 182–183
- RM (resource management) cells, 398
- RNB (rearrangeably nonblocking) switches, 582–585
- roots in Dijkstra's algorithm, 171–172
- routers, 64. *See also* switches and switching
  - border gateways, 172
  - interior and exterior, 172
  - Internet, 185
  - IP protocol, 160–163, 167–169
  - in label switching, 192
  - mobile IP, 175
  - multicast, 173–174
  - network layer, 110
  - TCP, 189

- routes, 63  
routing  
    Bellman-Ford algorithm for, 169–170  
    Border Gateway Protocol, 172–173  
    circuit-switched networks, 208  
    datagram circuits, 65–67  
    Dijkstra's algorithm for, 171–172  
    DWDM, 224  
    Ethernet networks, 121–125  
    factors, 366  
    IP protocols, 159, 167–173  
    multicast IP, 173–174  
    network layer, 109–110  
    with optical cross-connects, 559–560  
    optimizing. *See* routing optimization  
    policy-based, 195–196  
    wireless networks, 339–340, 365
- Routing Control Channels, 273
- routing header, 177
- routing optimization, 374
- circuit-switched networks
    - admission control, 378
    - dynamic alternate routing, 375
    - metastability and trunk reservations, 375–377
    - separable routing, 377
    - static routing, 374–375
  - datagram networks
    - Bellman-Ford algorithm, 384–385
    - distributed-gradient algorithm, 385–387
    - dynamic routing, 383–384
    - static routing, 381–383
  - routing protocols, new designs, 92
  - routing tables, searching, 591–593
  - routing tags, 589–590
- RS (Reed-Solomon) code, 70
- RS-232-C standard, 10–11
- RST flag, 179
- RSVP (Resource Reservation Protocol), 193–196
- RTP (real-time protocol), 194
- S —
- SA (source address), 118
- sampling, 21–22
- SAR (segmentation and reassembly)  
    sublayer, 282–283
- satellite networks, 309, 352–353
- scalability
- global multimedia networks, 637–639
  - MPEG standard, 250–251
- scale, economies of, 24–25
- Internet, 184
  - transmissions, 56–57
- scattering, 547
- SCM (subcarrier multiplexing), 554–555
- SCPs (service control points), 242
- SCR (sustained cell rate), 260, 262
- SDH (Synchronous Digital Hierarchy), 8, 212
- SDLC (Synchronous Data Link Control), 11–12
- SDUs (service data units), 282
- search engines, 4
- searching routing tables, 591–593
- section overhead (SOH)
- BISDN, 293
  - SONET, 216–219
- sections, 216–217
- security
- global multimedia networks, 621, 625, 627, 636–637, 641–642
  - importance of, 47
  - wireless networks, 314, 322, 341–342
- segmentation and reassembly (SAR)  
    sublayer, 282–283

- segments
  - IP protocol, 167–168
  - TCP, 178
- Selective Repeat Protocol (SRP), 77, 368
- self-routing, 593–596
- self-synchronization, 116
- semaphores, 83
- semi-orthogonal spreading codes, 332–333
- send command in FTP, 182
- sensitivity in optical receivers, 545
- separable routing, 377
- sequence number protection (SNP)
  - field, 283
- sequence numbers (SNs)
  - AAL, 283
  - ATM, 259
  - TCP, 178–179
- sequences with straight-line large deviations, 474–475
- serial ports, 10–11
- serial transmission protocols, 10–11
- service characteristics, 53–54
- service control points (SCPs), 242
- service data units (SDUs), 282
- service integration, 26–27
- service logic interpreter (SLI), 243
- service logic program (SLP), 243
- Service Plan Executive, 516
- Service Registry MIB module, 291
- service switching points (SSPs), 242–243
- session layer in OSI model, 111–112
- set-top boxes
  - CATV, 249
  - standards, 88–89
- Setup messages, 272
- setup time
  - circuit-switched networks, 208
  - switches, 576
- SF (start field), 284
- SFD (start-of-frame delimiter), 118
- shadow fading, 317–318
- Shannon capacity, 323–324
- shared access links, 208
- shared buffers, 605–607
  - multicasting in, 607
  - queuing analysis, 607–608
- shared memory method, 83
- shielded twisted pair (STP) wiring, 117
- shortcut models, 298–299
- shortest paths
  - Bellman-Ford algorithm for, 169–170
  - Dijkstra's algorithm for, 171–172
  - multicast IP, 173–174
- shot noise, 545
- shuffle networks, 563–565
- signal overlap, 229
- signal power, 545
- signal processing
  - global multimedia networks, 625
  - for intersymbol interference, 328–331
- signal propagation, 344
- signal-to-noise ratio (SNR)
  - digitization, 23–24
  - optical receivers, 545
- signal transfer points (STPs), 242
- signaling in ATM, 270–272, 292
- signaling system 7 (SS7), 242
- Simple Mail Transfer Protocol (SMTP), 182
- Simple Network Management Protocol (SNMP), 291–292
- simulated emissions, 544
- simultaneous transmissions, 117
- single-hop optical LANs, 561–563
- single-mode fibers, 553
- single resource pricing, 518–519
  - congestion prices, 523–526
  - cost recovery and optimum link capacity, 526–527

- single resource pricing (*cont.*)
  - usage-based, 520–523
- single switch circuit-switched networks, 443–446
- site rents, 493–494
- SLC (Subscriber Loop Carrier) system, 226
- SLI (service logic interpreter), 243
- slots in TDM, 59–60
- slotted ALOHA technique, 334
- SLP (service logic program), 243
- SM (statistical multiplexing), 12
  - ATM, 266–267, 416–419
  - operation, 59–62
- smart antennas, 331
- SMDS (Switched Multimegabit Data Service)
  - internetworking with, 145–147
  - operation, 142–145
  - usage charges, 509
- SMDS over ATM, 301
- SMT (station management) protocols, 132–133
- SMTP (Simple Mail Transfer Protocol), 182
- SNA (System Network Architecture), 369
- SNB (strictly nonblocking) switches, 582–585
- SNMP (Simple Network Management Protocol), 291–292
- SNP (sequence number protection) field, 283
- SNR (signal-to-noise ratio)
  - digitization, 23–24
  - optical receivers, 545
- SNs (sequence numbers)
  - AAL, 283
  - ATM, 259
  - TCP, 178–179
- social welfare optimum solutions, 522–523
- societal changes, 3–4
- sockets, 180–181
- soft capacity, 332–333
- software agents, 640
- SOH (section overhead)
  - BISDN, 293
  - SONET, 216–219
- solicitation messages, 176
- SONET (Synchronous Optical Network) standard, 90–91, 211–215
  - ATM fault management, 287
  - frame structure, 215–221
  - optical networking and future, 222–223
  - optical networks, 571
  - telephone networks, 8–9
- SONET-Light, 90
- source addresses
  - Ethernet packets, 118
  - IPv6, 176–177
  - packet switching, 67
- source/channel coding, 627, 638
- source port numbers, 178–179
- source service access points (SSAPs), 118
- sources, 51
- space-division switching, 580–582
- spanning tree routing
  - Bellman-Ford algorithm for, 170
  - Dijkstra's algorithm for, 171–172
  - Ethernet networks, 123–124
- SPE (synchronous payload envelope), 216–220
- spectral efficiency, 63
- spectral etiquette, 337
- speed. *See* performance
- splicing losses, 548–549
- spontaneous emissions, 544
- spread spectrum
  - with ALOHA technique, 334–335

- for intersymbol interference, 330–331  
wireless networks, 322, 350
- spreading codes, 332–333
- SRP (Selective Repeat Protocol), 77, 368
- SS7 (signaling system 7), 242
- SSAPs (source service access points), 118
- SSCs sublayer, 283
- SSPs (service switching points), 242–243
- standards
- global multimedia networks, 641
  - layered architectures, 82
  - wireless LANs, 350
- star architectures, 338–339
- start field (SF), 284
- start-of-frame delimiter (SFD), 118
- state space, 432–433
- state transition diagrams, 433–434
- stateless servers, 43–44
- states of Markov chains, 432
- static routing
- circuit-switched networks, 374–375
  - datagram networks, 381–383
- static tables, 590
- static wavelength assignment (SWA), 565–568
- station management (SMT) protocols, 132–133
- stations, 51
- statistical multiplexing (SM), 12
- ATM, 266–267, 416–419
  - operation, 59–62
- statistical procedures in ATM networks
- averaging rate fluctuations, 407–408
- buffering, 413–416
- vs. deterministic, 422–424
- GCRA in, 419–421
- Markov-modulated fluids, 405–406
- multiclass case, 419
- multiplexing, 408–410
- on-line estimation, 410–413
- statistical multiplexing and buffering, 416–419
- traffic models, 405
- traffic shaping, 421–422
- step-index fiber, 552–553
- stock purchases, 492
- stop command in rlogin, 183
- store-and-forward technique, 65
- Ethernet networks, 126
  - SDLC, 11–12
- STP (shielded twisted pair) wiring, 117
- STPs (signal transfer points), 242
- straight-line large deviations, 474–475
- stream mode in FTP, 182
- streams, audio and video, 42–43
- strictly nonblocking (SNB) switches, 582–585
- STS (Synchronous Transfer Signal) hierarchy
- SONET, 212
  - telephone networks, 8–9
- subcarrier multiplexing (SCM), 554–555
- subjective costs, 492
- subnet masks, 164–165
- subnetting, 164–165
- subscriber demand model, 497–501
- Subscriber Loop Carrier (SLC) system, 226
- subscriber loops, 57, 206
- supplementary ISDN services, 232–233
- sustained cell rate (SCR), 260, 262
- SWA (static wavelength assignment), 565–568
- Switched Multimegabit Data Service (SMDS)
- internetworking with, 145–147

- Switched Multimegabit Data Service (SMDS) (*cont.*)  
     operation, 142–145  
     usage charges, 509
- switches and switching, 51, 57, 575–576  
     ATM, 2, 266–267  
     DWDM, 224  
     Ethernet networks, 121, 125–127  
     improvements needed, 92  
     modular designs, 582–588  
     operation, 63–68  
     packet switching, 588–593. *See also* packet switching and packet-switched networks  
     performance measures, 576–579  
     queuing network model, 52  
     telephone networks, 7  
     time-division and space-division, 580–582
- SYN flag, 179–180
- synchronization  
     data link layer, 106  
     Ethernet packets, 118  
     ISDN, 234  
     physical layer, 104–105  
     10BASE-T networks, 116
- synchronization points, 112
- Synchronous Data Link Control (SDLC), 11–12
- Synchronous Digital Hierarchy (SDH), 8, 212
- Synchronous Optical Network. *See also* SONET (Synchronous Optical Network) standard  
     synchronous payload envelope (SPE), 216–220  
     synchronous traffic, 134–135
- Synchronous Transfer Signal (STS) hierarchy  
     SONET, 212  
     telephone networks, 8–9
- synchronous transmission standards, 11
- syntax resolution, 112–113
- System Network Architecture (SNA), 369
- T**
- T/TCP (Transactional TCP) protocol, 194
- tables  
     IP routing, 167–168  
     MAC addresses, 123, 126  
     packet switching, 590  
     searching, 591–593
- tags  
     packet switching, 589–590  
     switching, 192–193
- TAT (theoretical arrival time), 395
- TCP (Transmission Control Protocol), 178–183  
     improvement suggestions, 188–190  
     window adjustments, 186–190
- TCP-friendly protocols, 188
- TCP/IP networks, 155  
     FTP, 181–182  
     lock-in effect for, 490  
     performance, 186  
         IP improvement suggestions, 190  
         label switching, 191–193  
         queuing algorithms, 190–191  
         RSVP improvement suggestions, 193–196  
         TCP improvement suggestions, 188–190  
         window adjustments, 186–190  
     TCP and UDC, 178–183
- TD (transmission and propagation delay), 263
- TDM (time-division multiplexing)  
     ADSL, 235  
     FTTH, 226

- operation, 58–60  
SONET, 211–212  
TDMA (time-division multiple access)  
    technique, 332–333  
Telecommunications Act, 2, 30  
telecommunications industry, 1–2  
telecommuting, 54–55  
Teledesic satellite system, 353  
telephone networks  
    backbones, 50  
    cellular, 344–348  
    history, 6–10  
    POTS, 239–240  
    pricing schemes, 509  
    time-division switching, 580–581  
telephones, cordless, 348–349  
Telephony on PON (TPON) system,  
    227–229  
telephony servers, 196  
telepoints, 348–349  
teleservices in ISDN, 232–233  
television networks, cable. *See*  
    CATV (Community Antenna  
    Television) systems  
television signals, SNR for, 23–24  
10BASE-T networks, 115–116  
10BASE5 networks, 117  
terminal points, 216  
Textual Conventions MIB module, 291  
TFTP (Trivial File Transfer Protocol),  
    183  
theoretical arrival time (TAT), 395  
thermal noise, 545  
throughput of switches, 576  
THT (token holding time)  
    FDDI, 133–134  
    token ring networks, 129  
tiered services, 500–502  
time-division multiple access (TDMA)  
    technique, 332–333  
time-division multiplexing (TDM)  
    ADSL, 235  
    FTTH, 226  
    operation, 58–60  
    SONET, 211–212  
time-division switching, 580–582  
time-of-use charges, 508  
time reversible Markov chains, 451  
time scales, 367–368  
time to live, 111  
    IP packets, 163  
    routing tables, 168  
timed-token protocols, 133–135  
timers  
    multicast IP, 173  
    packet transmissions, 70  
timing out, 70–71  
token holding time (THT)  
    FDDI, 133–134  
    token ring networks, 129  
token ring access method, 14–15  
token ring networks, 127–128  
    LLC, 130  
    MAC, 128–130  
    physical layer, 128  
token rotation time timer (TTRT),  
    133–135  
top-level domains, 166  
topology state parameters, 273  
TOS (type of service), 163  
TPON (Telephony on PON) system,  
    227–229  
traceroute service, 169  
tracking systems, 354  
traffic characterization, 44–45  
    constant bit rate, 45–46  
    messages, 46–47  
    variable bit rate, 46  
traffic class field, 176  
traffic control and statistics  
    ATM, 290  
    global multimedia networks, 624  
traffic increase in high-performance  
    networks, 49–50

- traffic models  
     ATM networks, 405  
     averaging rate fluctuations, 407–408  
     buffering, 413–416  
     GCRA in, 419–421  
     Markov-modulated fluids, 405–406  
     multiclass case, 419  
     multiplexing, 408–410  
     on-line estimation, 410–413  
     statistical multiplexing and  
         buffering, 416–419  
         traffic shaping, 421–422  
     traffic policing, 285  
     traffic shaping, 366, 421–422  
     trailers, 11  
     transaction costs, 491–492  
     Transactional TCP (T/TCP) protocol,  
         194  
     transfer command in FTP, 182  
     transfer rates, 51  
     transferring files, 181–182  
     transition diagrams, 444–445  
     transition probability matrices,  
         433–435  
     translators, 196  
     transmission and propagation delay  
         (TD), 263  
     Transmission Control Protocol (TCP),  
         178–183  
         improvement suggestions, 188–190  
         window adjustments, 186–190  
     transmission delays, 53  
     transmission rates, 186–188  
     transmissions, economies of scale in,  
         56–57  
     transmitters  
         data link layer, 105–106  
         optical LANs, 561–562  
         optical links, 543–544  
         physical layer, 104  
     transparency  
         global multimedia networks, 637  
     WDM systems, 557  
     transparent routing, 122–123  
     transport layer in OSI model, 110–111  
     transport protocols, 92  
     transport service access points  
         (TSAPs), 110–111  
     travel agents, 492  
     tree searches, 592  
     Trivial File Transfer Protocol (TFTP),  
         183  
     trunk reservation, 376–377  
     trunks, 57–58  
     TSAPs (transport service access  
         points), 110–111  
     TTRT (token rotation time timer),  
         133–135  
     tunnels  
         IPv6, 176  
         multicast IP, 174  
     Turbo codes, 326  
     Turbo equalization, 329  
     twisted pair wiring  
         10BASE-T networks, 117  
         10BASE-T networks, 115–116  
     Type 1 traffic in AAL, 283  
     Type 2 traffic in AAL, 283–284  
     Type 3/4 traffic in AAL, 284  
     Type 5 traffic in AAL, 285  
     type of service (TOS), 163
- ■ ■ U
- UBR (unspecified bit rate) services  
     ATM, 260–262  
     GCRA, 395, 398  
     UDC (User Data Protocol), 178–183  
     UNI (user-network interface),  
         270–271, 277  
     unique invariant distributions,  
         435–436  
     Universal Reference Locators (URLs),  
         167

- unshielded twisted pair (UTP) wiring  
  100BASE-T networks, 117  
  10BASE-T networks, 115–116
- unspecified bit rate (UBR) services  
  ATM, 260–262  
  GCRA, 395, 398
- UPC (usage parameter control), 268
- URG flag, 179
- URLs (Universal Reference Locators), 167
- usage-based pricing, 520–523
- usage charges, 507–508
- usage parameter control (UPC), 268
- User Data Protocol (UDC), 178–183
- user experience in billing systems, 512–513
- user interaction requests, 243–244
- user-network interface (UNI), 270–271, 277
- user/network signaling, 292
- user planes, 286
- users, 51
- utility functions, 520
- utilization  
  circuit-switched networks, 209  
  multiplexing channels, 59
- UTP (unshielded twisted pair) wiring  
  100BASE-T networks, 117  
  10BASE-T networks, 115–116
- V —
- V.34 modems, 238–239
- V.90 modems, 238–239
- variable bit rate (VBR)  
  acceptable, 46  
  GCRA, 395, 397
- variable bit rate-non-real time (VBR-NRT) services, 260, 262
- variable bit rate-real time (VBR-RT) services, 260, 262
- variable-length subnet masks, 165
- variable quality in billing systems, 513–515
- VBR (variable bit rate)  
  acceptable, 46  
  GCRA, 395, 397
- VBR-NRT (variable bit rate-non-real time) services, 260, 262
- VBR-RT (variable bit rate-real time) services, 260, 262
- VCIs (virtual channel identifiers), 259  
  ATM, 277–282  
  packet switching, 589–591
- VDSL (very high data rate DSL), 237
- vehicle tracking systems, 354
- version field, 176
- vertical handoffs, 634
- very high data rate DSL (VDSL), 237
- vibrational absorption, 547
- video applications  
  coding for, 631–632, 637–638  
  global multimedia networks, 626  
  program distribution, 31
- video dial-tone, 249
- video on demand, 249
- video streams, 42–43
- videoconferences, 43
- virtual channel identifiers (VCIs), 259  
  ATM, 277–282  
  packet switching, 589–591
- virtual channels, 259
- virtual circuit networks  
  packet, 66–68  
  performance, 258
- virtual circuits, 144
- virtual corporations, 3
- virtual LANs (VLANs), 127
- virtual locations, 491–492
- virtual path identifiers (VPIS), 259, 277–282
- virtual private networks (VPNs), 278
- virtual routing, 300
- VLANs (virtual LANs), 127

- voice over packets, 43  
VPIs (virtual path identifiers), 259, 277–282  
VPNs (virtual private networks), 278
- W**
- wave-division multiplexing (WDM)  
systems, 230–231, 548, 556–557  
wave-mixing, 559  
wavelength assignment, 565–568  
wavelength continuity, 560, 566  
wavelength conversion  
DWDM, 224  
with optical cross-connects, 558  
wavelength routing  
DWDM, 224  
with optical cross-connects, 559–560  
wavelength-selective optical cross connects (WSXC), 224  
wavelength-selective switches, 557  
WDM (wave-division multiplexing)  
systems, 230–231, 548, 556–557  
Web sites, 4, 42  
weighted RED mechanism, 391  
weighted round-robin strategy, 62  
well-known ports, 110–111  
wide area wireless data services, 351  
window adjustments, 186–190  
window congestion control, 388–391  
wireless networks, 305–306  
ad hoc, 355–356  
ATM, 354–355  
capacity, 323–324  
CATV, 246–247  
cellular systems, 344–348  
channel access, 331–332  
multiple access, 332–334  
random access, 334–337  
spectral etiquette, 337  
channels in, 315–316
- cordless phones, 348–349  
design  
architecture, 337–339  
internetworking, 341  
mobility management, 339–340  
new paradigm, 342–343  
reliability, 340–341  
security, 341–342  
Doppler frequency shift, 321  
Ethernet, 117  
fixed wireless access, 357–358  
future, 30, 309–312  
high-speed digital cellular, 357  
history, 19–21, 306–309  
HomeRF and Bluetooth, 358  
IMT-2000, 356–357  
infrared vs. radio, 322–323  
interference, 321–322  
LANs, 349–351, 354–355  
link level design, 324  
channel coding and link layer  
retransmission, 325–326  
for flat-fading, 326–328  
for intersymbol interference, 328–331  
modulation techniques, 324–325  
multipath flat-fading and  
intersymbol interference, 318–321  
paging systems, 352  
path loss, 316–317  
remote sensing applications, 353–354  
satellite, 352–353  
shadow fading, 317–318  
technical challenges, 312–315  
wide area, 351
- wiring  
100BASE-T networks, 117  
10BASE-T networks, 115–116
- World Trade Organization  
agreement, 2

World Wide Web, 4, 42, 156  
WSXC (wavelength-selective optical cross connects), 224

**— Z**

zero forcing, 328–329

**— X**

X.25 protocol, 138–139